La Logica nell'era delle reti neurali

Strumento insostituibile di certificazione, governance e ragionamento rigoroso

Monografia sul ruolo del formalismo logico nell'intelligenza artificiale contemporanea

INDICE DELLE PARTI

Parte I Il Grande Scisma e le sue Conseguenze Intellettuali

Parte II I Limiti della Logica: Precisi, Noti e Gestibili

Parte III I Limiti Strutturali del Connessionismo

Parte IV I Fondamenti Matematici delle Reti Neurali Profonde

Parte V La Logica come Strumento di Certificazione e Verifica

Parte VI Logica Probabilistica, Teoria dell'Informazione e Architetture Neuro-Simboliche

Parte VII La Logica come Linguaggio della Governance AI

Parte VIII Questioni Aperte e Frontiere della Ricerca

Parte IX La Filosofia dei Due Paradigmi

Parte X Conclusioni

Parte App. Appendici e Bibliografia

Silvano Salvador

Prefazione: Una domanda urgente

Il successo spettacolare dei grandi modelli linguistici e delle reti neurali profonde ha alimentato, negli ultimi anni, una narrazione di obsolescenza: la logica simbolica, il ragionamento formale, i sistemi a regole esplicite sarebbero stati definitivamente superati da architetture che imparano dai dati senza bisogno di conoscenza codificata. La domanda che questa monografia si propone di affrontare — con la precisione tecnica che l'argomento richiede — è se questa narrazione sia corretta.

La risposta che si difenderà nelle pagine seguenti è negativa, e per ragioni strutturali piuttosto che di preferenza metodologica. La logica matematica non è diventata irrilevante nell'era del deep learning: essa ha mutato il proprio ruolo, spostandosi dal centro dell'architettura dei sistemi AI alla loro periferia normativa e certificativa. Ma quella periferia è diventata, con la pervasività dei sistemi AI nelle infrastrutture critiche, il punto strategicamente più importante dell'intero panorama tecnologico.

Tre argomenti strutturali sostengono questa tesi. Primo: i sistemi neurali hanno limiti computazionali che non sono superabili con più dati o più parametri — limiti nella certificazione del comportamento, nell'assenza di garanzie out-of-distribution, nell'incapacità di ragionamento causale e controfattuale — e questi limiti richiedono l'integrazione di strutture formali. Secondo: la logica matematica è diventata il linguaggio con cui si formulano le specifiche di comportamento, si verificano le proprietà di sicurezza e si certificano i requisiti di equità — funzioni che nessun sistema di apprendimento può svolgere su sé stesso. Terzo: l'architettura neuro-simbolica — in cui modelli linguistici suggeriscono e proof

Silvano Salvador

assistants certificano — è la direzione tecnologica più promettente per sistemi AI che combinino versatilità statistica e garanzie formali.

La monografia è organizzata in dieci parti. Le Parti I e II tracciano la storia del confronto tra i due paradigmi e analizzano i limiti della logica formale nella loro corretta interpretazione. Le Parti III e IV analizzano simmetricamente i limiti strutturali del connessionismo e i fondamenti matematici delle reti neurali. Le Parti V, VI e VII sviluppano la tesi principale: il ruolo nuovo e insostituibile della logica come strumento di certificazione, architettura neuro-simbolica e linguaggio della governance AI. Le Parti VIII, IX e X affrontano le questioni aperte, le fondamenta filosofiche e le conclusioni.

PARTE I

Il Grande Scisma e le sue Conseguenze Intellettuali

1. Due paradigmi, una domanda permanente

1.1 La domanda originaria e la sua urgenza contemporanea

Ogni riflessione seria sull'intelligenza artificiale deve prima interrogarsi su che cosa si intenda per intelligenza tout court — interrogativo che la filosofia della mente ha affrontato con strumenti progressivamente più sofisticati, senza tuttavia pervenire a una risposta unanime. Ma nell'anno 2024, questa domanda ha assunto un'urgenza pratica che va al di là delle dispute accademiche: i sistemi AI gestiscono infrastrutture critiche, partecipano a decisioni cliniche, generano contenuti che influenzano miliardi di persone, e operano in domini dove un errore può avere

Silvano Salvador

conseguenze irreversibili. In questo contesto, la questione di quale struttura formale governa — o dovrebbe governare — il loro comportamento non è filosofica ma ingegneristica e normativa.

La domanda fondante può essere formulata nel modo seguente: il ragionamento razionale — umano o artificiale — è riducibile a manipolazione di simboli secondo regole esplicite, oppure esso emerge da processi distribuiti, subsimbolici, il cui substrato computazionale è radicalmente diverso da quello di un sistema formale? La risposta che si dà a questa domanda non è indifferente rispetto ai metodi che si adottano per costruire macchine intelligenti, né rispetto agli strumenti che si utilizzano per certificarne la correttezza e governarne l'impatto sociale.

Questo saggio sostiene una tesi precisa: il connessionismo ha vinto la battaglia dei compiti percettivi, linguistici e statistici. Ma la logica simbolica ha conquistato, in risposta, un territorio nuovo e strategicamente più importante: la certificazione, la governance e il ragionamento rigoroso sui sistemi AI stessi. Lungi dall'essere obsoleta, la logica matematica è diventata il linguaggio con cui l'umanità deve parlare ai propri sistemi AI più potenti — per vincolarli, per verificarli, per comprenderli.

1.2 L'eredità leibniziana e il progetto del calcolo universale

Le radici della tradizione simbolica nell'intelligenza artificiale affondano nel progetto intellettuale di Gottfried Wilhelm Leibniz (1646–1716), che concepì la possibilità di una characteristica universalis — un linguaggio formale abbastanza ricco da poter esprimere ogni contenuto del pensiero umano — e di un calculus ratiocinator che, operando su tale linguaggio, avrebbe consentito di risolvere ogni disputa intellettuale mediante il

Silvano Salvador

computo. La celebre battuta leibniziana calculemus — «calcoliamo» — riassume con eloquenza il programma: ridurre l'argomentazione a un algoritmo deterministico, il ragionamento a un calcolo.

Questo sogno trovò la propria concretizzazione matematica rigorosa nel tardo Ottocento, con il lavoro di George Boole, Gottlob Frege, Giuseppe Peano e infine Bertrand Russell e Alfred North Whitehead. Boole dimostrò che le operazioni logiche elementari — congiunzione, disgiunzione, negazione — potevano essere trattate algebricamente, aprendo la via a un calcolo proposizionale formale. Frege, con la Begriffsschrift del 1879, costruì il primo sistema assiomatico della logica del primo ordine, abbastanza potente da fondare l'intera aritmetica su basi puramente logiche.

Il progetto logicista, anche laddove non si traduce in un impegno ontologico pieno circa la riduzione della matematica alla logica, lascia in eredità all'intelligenza artificiale simbolica due assunti strutturali: primo, che il contenuto del pensiero è rappresentabile mediante strutture discrete e composizionali; secondo, che le trasformazioni legittime su tali strutture sono descrivibili mediante regole finitamente specificabili. Questi assunti costituiranno il nucleo della Physical Symbol System Hypothesis.

1.3 Il sistema formale: struttura e potere espressivo

Un sistema formale S è una quadrupla $\langle \Sigma, F, A, R \rangle$ dove Σ è un alfabeto finito o numerabile di simboli primitivi, $F \subseteq \Sigma^*$ è l'insieme delle formule ben formate (wff) definito induttivamente, $A \subseteq F$ è l'insieme degli assiomi, e R è un insieme finito di regole di inferenza. Una derivazione di φ da Γ è una sequenza finita $\varphi_1, ..., \varphi = \varphi$ dove ogni elemento è un

Silvano Salvador

assioma, un elemento di Γ, oppure ottenibile da elementi precedenti mediante una regola.

La logica del primo ordine (FOL) è lo strumento formale principale dell'AI simbolica. Il suo alfabeto comprende variabili individuali x, y, z, ...; simboli di costante a, b, c, ...; simboli di funzione e predicato di arità k; connettivi logici $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$; quantificatori $\forall$, $\exists$. L'assiomatizzazione di Hilbert-Ackermann include tra i suoi schemi fondamentali:

(A1) $\varphi \rightarrow (\psi \rightarrow \varphi)$

(A2) $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$

(A3) $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$

(A4) $\forall x\, \varphi(x) \rightarrow \varphi(t)$ [per ogni termine t libero per x in φ]

(A5) $\forall x\, (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \forall x\, \psi)$ [con x non libera in φ]

Il Teorema di Completezza di Gödel (1930) afferma che la logica del primo ordine è corretta e completa: per ogni insieme di enunciati Γ e ogni enunciato φ, $\Gamma \vdash \varphi$ se e solo se $\Gamma \vDash \varphi$. Questo risultato — distinto e opposto, in direzione, dai successivi teoremi di incompletezza — garantisce che ogni verità logica di primo ordine è dimostrabile nel sistema formale standard.

1.4 Newell, Simon e la Physical Symbol System Hypothesis

Nel 1975, ricevendo il Premio Turing, Allen Newell e Herbert Simon formularono il manifesto del paradigma simbolico. La Physical Symbol System Hypothesis (PSSH) afferma che un sistema fisico capace di manipolare simboli secondo regole strutturali è condizione necessaria e sufficiente per l'intelligenza generale. La tesi è forte in entrambe le

Silvano Salvador

direzioni: necessaria, nel senso che ogni sistema intelligente, incluso il cervello umano, è un sistema a simboli fisici; sufficiente, nel senso che un sistema meccanico adeguatamente programmato per manipolare simboli è già, in virtù di ciò solo, intelligente.

La General Problem Solver di Newell e Simon, il sistema LISP, i sistemi esperti degli anni Settanta e Ottanta — MYCIN per la diagnosi medica, DENDRAL per la chimica organica, XCON per la configurazione di sistemi informatici — incarnano questa visione con precisione quasi paradigmatica. Il successo di queste applicazioni nei domini ristretti e ben delimitati sembrò, per un certo periodo, confermare la correttezza dell'approccio. MYCIN, in particolare, mostrava prestazioni diagnostiche comparabili a quelle di specialisti clinici umani in un dominio strettamente delimitato, codificando circa 600 regole di produzione della forma SE-ALLORA.

La struttura inferenziale di MYCIN

Le regole di MYCIN avevano la forma: SE [organismo] ha gram-stain = gram-positive AND morphology = coccus AND [organismo] cresce in catene ALLORA c'è suggestive evidence (0.7) che l'identità dell'organismo sia streptococcus.

Il sistema combinava i fattori di certezza (CF) mediante: CF(A AND B) = min(CF(A), CF(B)); CF(A OR B) = max(CF(A), CF(B)); CF(NOT A) = −CF(A). Questa era logica sfumata prima che il termine fosse formalizzato.

Il sistema funzionava bene perché il dominio era chiuso: tutte le entità rilevanti (batteri, antibiotici, parametri clinici) erano pre-enumerati. Il frame problem non si manifestava.

1.5 La critica connessionista: biologia e Perceptron

Il paradigma simbolico conteneva in sé una tensione irrisolta che i suoi avversari non tardarono a mettere in luce. Il cervello umano non assomiglia, né nella struttura né nella dinamica, a un sistema formale.

Silvano Salvador

Esso è costituito da circa $8{,}6 \times 10^{10}$ neuroni, ciascuno connesso in media a 7×10^{3} sinapsi, per un totale di circa $1{,}5 \times 10^{14}$ connessioni sinaptiche. Il suo funzionamento è distribuito, parallelo, robusto alla degradazione parziale, intrinsecamente probabilistico.

Warren McCulloch e Walter Pitts, nel celebre articolo del 1943, dimostrarono che reti di neuroni formali, concepiti come sogliatori binari, erano in grado di computare qualsiasi funzione booleana — aprendo paradossalmente la tradizione connessionista proprio con uno strumento della logica matematica. Il Perceptron di Frank Rosenblatt (1958) introdusse l'idea cruciale di pesi adattivi: un neurone formale calcola $f(x) = \sigma(w^{T}x + b)$, dove σ è una funzione di soglia, e i pesi w vengono aggiornati mediante una regola di apprendimento derivata dall'errore di classificazione.

Il limite fondamentale, dimostrato rigorosamente da Minsky e Papert nel 1969, è che i percettroni a singolo strato non possono computare funzioni non linearmente separabili. La funzione XOR — $\varphi: \{0,1\}^{2} \to \{0,1\}$ con $\varphi(0,0) = \varphi(1,1) = 0$ e $\varphi(0,1) = \varphi(1,0) = 1$ — non è linearmente separabile nel piano: non esiste una retta che separi i punti positivi da quelli negativi. La dimostrazione di Minsky e Papert decretò un lungo inverno del connessionismo.

1.6 La rinascita: dalla retropropagazione ai Transformer

Il risveglio del connessionismo avvenne attraverso una serie di contributi convergenti. La svolta decisiva giunse con la diffusione dell'algoritmo di retropropagazione dell'errore da parte di Rumelhart, Hinton e Williams (1986), che consentiva di addestrare reti con molteplici strati nascosti mediante discesa del gradiente sull'errore. Hinton, in particolare, fu il più tenace assertore della tesi connessionista: il cervello non deduce in senso

Silvano Salvador

formale, esso inferisce probabilisticamente, comprimendo informazioni in rappresentazioni distribuite che non corrispondono ad alcun simbolo atomico.

A partire dal 2012, con la rete AlexNet di Krizhevsky, Sutskever e Hinton, e poi con la progressione esponenziale delle architetture Transformer, il connessionismo ha ottenuto successi su compiti — visione artificiale, comprensione del linguaggio naturale, sintesi proteica, generazione di testi e immagini — che il paradigma simbolico non aveva mai nemmeno sfiorato. I grandi modelli linguistici (LLM) come GPT-4, Claude, Gemini mostrano capacità emergenti sorprendenti: ragionamento su più passi, traduzione zero-shot, soluzione di problemi matematici.

La domanda che rimane aperta

Sarebbe affrettato concludere da questo trionfo che la logica matematica sia diventata irrilevante. La questione è più sottile. Il dominio applicativo in cui i sistemi neurali eccellono — approssimazione di funzioni, classificazione statistica, generazione — è radicalmente diverso dal dominio in cui la logica formale è insuperabile: la verifica formale, la dimostrazione automatica di teoremi, il ragionamento matematico rigoroso, la garanzia di correttezza certificabile.

Questa monografia argomenta che la complementarità è strutturale, non accidentale: emerge dalla geometria matematica dei due paradigmi, non da preferenze metodologiche. Comprenderla richiede di analizzare con precisione i limiti di entrambi.

PARTE II

I Limiti della Logica: Precisi, Noti e Gestibili

Silvano Salvador

2. I teoremi di incompletezza di Gödel e la loro interpretazione corretta

2.1 Il programma di Hilbert: l'ambizione vanificata

Per apprezzare la portata rivoluzionaria dei teoremi di incompletezza di Kurt Gödel (1931), è necessario ricostruire l'ambizione del programma che essi vanificarono. David Hilbert, al Congresso Internazionale dei Matematici di Parigi nel 1900, aveva formulato i suoi celebri ventitré problemi. Nel 1928, egli radicalizzò questa ambizione con il Hilbert program: dimostrare, con metodi finitari, la consistenza di ogni sistema formale abbastanza potente da fondare l'aritmetica, e dimostrare la completezza — che ogni proposizione matematicamente ben formulata fosse o dimostrabile o confutabile.

L'assunzione tacita era che la matematica fosse, in linea di principio, decidibile: dato un enunciato φ e un sistema T, esiste una procedura meccanica che determina se $T \vdash \varphi$ oppure $T \vdash \neg\varphi$. Questa era la forma forte del programma. Poi, nel 1931, Gödel pubblicò «Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I» — un articolo che avrebbe cambiato per sempre l'immagine che la matematica aveva di sé stessa.

2.2 La costruzione dell'enunciato di Gödel: diagonalizzazione

La tecnica fondamentale dei teoremi di incompletezza è la numerazione di Gödel: a ogni simbolo dell'alfabeto di T si assegna un numero naturale, e a ogni formula, sequenza di formule e dimostrazione si assegna un numero calcolato mediante una codifica composizionale basata sulla fattorizzazione in numeri primi. Questo permette di parlare

Silvano Salvador

di enunciati e dimostrazioni dentro il sistema T, purché esso sia abbastanza espressivo da quantificare sui numeri naturali.

Si definisce il predicato Prov_T(x, y) come «y è il numero di Gödel di una dimostrazione in T dell'enunciato con numero di Gödel x». Il predicato di dimostrabilità è Bew_T(x) ≡ ∃y Prov_T(x, y). L'enunciato di Gödel è costruito mediante il Lemma Diagonale: per ogni formula φ(v) con una variabile libera, esiste un enunciato γ tale che:

$$T \vdash \gamma \leftrightarrow \varphi(\ulcorner \gamma \urcorner)$$

dove $\ulcorner \gamma \urcorner$ è il numerale corrispondente al numero di Gödel di γ. Scegliendo φ(v) ≡ ¬Bew_T(v) si ottiene l'enunciato G_T che «dice» di sé stesso di essere indimostrabile in T:

$$G_T \equiv \neg \mathrm{Bew}_T(\ulcorner G_T \urcorner)$$

Primo Teorema (Gödel 1931, Rosser 1936): Se T è ricorsivamente assiomatizzabile e consistente e soddisfa le condizioni di rappresentabilità della funzione di Gödel, allora T è incompleto: esistono G_T tale che $\nvdash$_T G_T e $\nvdash$_T ¬G_T. La variante di Rosser indebolisce l'ipotesi da ω-consistenza a mera consistenza.

Secondo Teorema (Gödel 1931): Nelle stesse ipotesi, T non può dimostrare la propria consistenza: $\nvdash$_T Con(T), dove Con(T) ≡ ¬Bew_T($\ulcorner \bot \urcorner$).

2.3 Il Teorema di Löb e la logica della provability

Un corollario di importanza straordinaria è il Teorema di Löb (1955): per ogni enunciato φ,

$$T \vdash \mathrm{Bew}_T(\ulcorner \varphi \urcorner) \rightarrow \varphi \quad \text{se e solo se} \quad T \vdash \varphi$$

Silvano Salvador

In altri termini: l'unico enunciato φ per cui T può dimostrare «se φ è dimostrabile allora φ è vero» è un enunciato che T dimostra già incondizionatamente. La Logica della Provability (GL, per Gödel-Löb), sviluppata da Solovay (1976), è una logica modale in cui □ interpreta Bew_T. I suoi assiomi caratteristici sono:

(GL1) $\Box\varphi \rightarrow \varphi$ [se è dimostrabile, è vero — ma solo per teoremi]

(GL2) $\Box(\Box\varphi \rightarrow \varphi) \rightarrow \Box\varphi$ [Löb: l'autoreferenza è sterile]

(GL3) $\Box\varphi \rightarrow \Box\Box\varphi$ [introspection]

Il teorema di completezza di Solovay afferma che GL cattura esattamente le leggi modali valide per il predicato di dimostrabilità dell'Aritmetica di Peano — un risultato che unifica logica modale, teoria della dimostrazione e aritmetica.

2.4 Perché i teoremi di Gödel non confutano la logica applicata

I teoremi di incompletezza sono spesso usati in modo inappropriato per argomentare contro la rilevanza della logica formale. Occorre una distinzione precisa. I teoremi stabiliscono: (a) che nessun sistema formale coerente e sufficientemente espressivo può essere completo per l'aritmetica; (b) che nessun tale sistema può dimostrare la propria consistenza.

Questi risultati hanno conseguenze precise per la fondazione della matematica. Ma per la maggior parte delle applicazioni ingegneristiche della logica formale, essi sono sostanzialmente irrilevanti. La gerarchia aritmetica classifica i problemi in base alla complessità delle formule necessarie per definirli. Gli insiemi $\Delta_1 = \Sigma_1 \cap \Pi_1$ coincidono con i problemi decidibili. Il model checking per la logica temporale CTL è

Silvano Salvador

P-completo. La soddisfacibilità proposizionale (SAT) è NP-completa ma affrontabile nella pratica da solver moderni. La verifica di protocolli su sistemi finiti è decidibile.

Il punto cruciale che la narrativa popolare ignora

Nessun ingegnere del software che usa Coq, Lean, o Z3 per verificare un protocollo crittografico si trova mai a fronteggiare un enunciato gödelianamente indecidibile. I sistemi che si verificano nella pratica — compilatori, kernel di sistemi operativi, protocolli di rete — hanno proprietà che si collocano nei frammenti decidibili della logica, per i quali i teoremi di incompletezza non si applicano.

Dire che 'la logica ha limiti gödeliani' come argomento contro la verifica formale è come dire che 'la fisica classica è limitata dalla meccanica quantistica' come argomento contro l'utilità dell'ingegneria strutturale. I limiti sono reali ma si manifestano in regimi radicalmente diversi da quelli ingegneristici.

2.5 L'argomento di Penrose e le sue confutazioni

La tesi di Lucas (1961) e Penrose (1989, 1994) sostiene che i teoremi di incompletezza dimostrano l'impossibilità di un'AI forte: nessun sistema formale può catturare la piena potenza del ragionamento matematico umano, poiché un agente conscio può sempre «vedere» la verità di G_T per il sistema T entro cui opera, mentre T non può dimostrarlo.

La risposta standard — articolata da Dennett, Boolos, Feferman e Shapiro — è multipla. In primo luogo, non è dimostrato che la mente umana sia un sistema formale consistente: gli esseri umani commettono errori logici sistematici, non sono ω-consistenti, e non è affatto evidente che possano «vedere» la verità di Gödel in modo affidabile e non circolare. In secondo luogo, anche se la mente umana fosse più potente di ogni singolo sistema formale, ciò non implicherebbe non-computabilità: una gerarchia transfinita di sistemi formali, ciascuno più forte del precedente, potrebbe simulare tale potenza. In terzo luogo,

Silvano Salvador

l'argomentazione presuppone che la mente abbia accesso privilegiato alla propria struttura formale, il che è epistemologicamente controverso.

3. Decidibilità, calcolabilità e la mappa dei limiti computazionali

3.1 La macchina di Turing e la tesi di Church-Turing

La teoria della calcolabilità, sviluppata in modo indipendente e convergente da Alonzo Church (λ-calcolo, 1936), Alan Turing (macchine di Turing, 1936) e Emil Post (sistemi di riscrittura, 1936), fornisce la formalizzazione matematica del concetto di «procedura meccanica». Una macchina di Turing M è una settupla $\langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ dove Q è un insieme finito di stati, Γ è l'alfabeto del nastro, $b \in \Gamma$ il simbolo bianco, $\Sigma \subseteq \Gamma \setminus \{b\}$ l'alfabeto di input, $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ la funzione di transizione parziale, q_0 lo stato iniziale, $F \subseteq Q$ l'insieme degli stati di accettazione.

La Tesi di Church-Turing afferma che ogni funzione «effettivamente calcolabile» nel senso intuitivo è calcolabile da una macchina di Turing. Questo non è un teorema dimostrabile ma è sostenuto dall'equivalenza dimostrata tra tutti i formalismi proposti: λ-calcolo, macchine di Turing, funzioni ricorsive parziali, sistemi di Post, macchine RAM sono tutti equivalenti in termini di potere computazionale.

3.2 Il Problema della Fermata: la prima e più fondamentale indecidibilità

Silvano Salvador

Il risultato di insolubilità più fondamentale della teoria della calcolabilità è la non-decidibilità del Problema della Fermata: non esiste una macchina di Turing che, su ogni coppia ⟨M, x⟩, risponda correttamente se M termina su x. La dimostrazione è per diagonalizzazione:

Si supponga per assurdo che esista una macchina H che decide HP. Si costruisce D che su input ⟨M⟩ simula H su ⟨M, ⟨M⟩⟩ e fa il contrario: se H accetta, D entra in loop; se H rifiuta, D termina. Si chiede cosa fa D su ⟨D⟩: in entrambi i casi si ottiene una contraddizione. □

Questo risultato ha implicazioni immediate: l'Entscheidungsproblem — decidere la validità di una formula di FOL — è anch'esso non decidibile (Church, 1936; Turing, 1936). La validità logica in FOL è semidecidibile: esiste un procedimento che enumera tutte le tautologie, ma non uno che risponde in tempo finito per ogni formula.

3.3 Il Teorema di Tarski e l'indefinibilità della verità

Teorema di Tarski (1933): Non esiste una formula Truth(x) nel linguaggio dell'aritmetica del primo ordine tale che, per ogni enunciato φ,

$$\mathbb{N} \vDash \text{Truth}(\ulcorner \varphi \urcorner) \leftrightarrow \varphi$$

La dimostrazione è analoga al paradosso del mentitore: se Truth esistesse, si potrebbe costruire l'enunciato $\lambda \equiv \neg\text{Truth}(\ulcorner \lambda \urcorner)$, portando a una contraddizione. Il teorema di Tarski stabilisce che la verità in $\mathbb{N}$ non è definibile nel linguaggio dell'aritmetica stessa — un risultato più forte dell'incompletezza gödeliana.

3.4 Il Teorema di Rice: l'insolubilità delle proprietà semantiche

Silvano Salvador

Teorema di Rice (1953): Ogni proprietà semantica non banale dei programmi — cioè ogni proprietà che dipende dal comportamento e non è vera di tutti o di nessun programma — è non decidibile.

Formalmente: sia P una proprietà tale che l'insieme I_P = {⟨M⟩ : M ha la proprietà P} non sia né ∅ né l'insieme di tutte le macchine di Turing. Allora I_P non è decidibile. La dimostrazione si riduce al Problema della Fermata per via di riduzione.

Il Teorema di Rice implica che nessun compilatore o analizzatore statico può decidere in generale: «questo programma termina sempre», «questo programma non ha buffer overflow», «questo programma è corretto rispetto alla sua specifica». Questi risultati non invalidano la verifica formale: essi definiscono i suoi confini e motivano l'uso di frammenti decidibili — linguaggi con espressività limitata, proprietà sufficientemente semplici, sistemi finiti.

4. Complessità computazionale del ragionamento logico

4.1 La gerarchia di complessità

La teoria della complessità fornisce un secondo strato di vincoli: non ciò che è computabile in linea di principio, ma ciò che è computabile efficientemente. Le classi fondamentali sono:

- P: problemi risolubili in tempo polinomiale deterministico;
- NP: problemi per cui una soluzione può essere verificata in tempo polinomiale;
- PSPACE: problemi risolubili in spazio polinomiale (P ⊆ NP ⊆ PSPACE);

Silvano Salvador

- EXPTIME: problemi risolubili in tempo esponenziale.

Problema logico	Complessità	Note pratiche
Soddisfacibilità proposizionale (SAT)	NP-completo (Cook, 1971)	Solver moderni (Z3, Glucose) scalano a milioni di variabili in casi pratici
Verifica di tautologia proposizionale	coNP-completo	Complemento di SAT
Ragionamento in logica modale K, T, S4, S5	PSPACE-completo	Decidibile; solver basati su tableau
Ragionamento in logica descrittiva ALC	EXPTIME-completo	Base formale delle ontologie OWL
Model checking CTL su sistemi finiti	P-completo	Usato industrialmente (SPIN, nuSMV)
Model checking CTL* su sistemi finiti	PSPACE-completo	Espressività maggiore, costo maggiore
Soddisfacibilità FOL (Entscheidungsproblem)	Semidecidibile	Teorema di Church-Turing (1936)
Validità FOL	Non ricorsivamente enumerabile	Complemento non semidecidibile

Questi risultati hanno implicazioni concrete per la progettazione di sistemi di AI simbolica: scegliere una logica troppo espressiva significa rinunciare all'efficienza computazionale; scegliere una logica troppo povera significa perdere capacità espressiva. Il compromesso tra espressività e trattabilità è uno dei temi tecnici più delicati dell'AI simbolica, e la ricerca in logiche trattabili (EL, DL-Lite, logiche Horn) mira precisamente a identificare frammenti che bilancino questi requisiti.

Silvano Salvador

4.2 La risoluzione di Robinson e l'automated theorem proving

Il contributo di John Alan Robinson (1965) con il principio di risoluzione trasformò la logica del primo ordine in uno strumento computazionale praticabile. La risoluzione è una regola di inferenza refutazionale: per dimostrare che $\Gamma \models \varphi$, si aggiunge $\neg\varphi$ all'insieme Γ e si cerca di derivare la clausola vuota $\square$. Il sistema è refutation-complete: se $\Gamma \cup \{\neg\varphi\}$ è insoddisfacibile, la risoluzione lo scoprirà in tempo finito.

La procedura richiede la conversione in forma normale congiuntiva (CNF) e la skolemizzazione. La regola di risoluzione afferma: date due clausole $C_1 = L \vee \alpha$ e $C_2 = \neg L' \vee \beta$, dove L e L' sono unificabili con sostituzione σ ($L\sigma = L'\sigma$), il risolvente è $(\alpha \vee \beta)\sigma$. L'algoritmo di unificazione di Robinson calcola il most general unifier (MGU) in tempo quasi lineare.

Questo fu il nucleo tecnico attorno a cui si costruì Prolog e la tradizione dell'automated theorem proving. I sistemi contemporanei — OTTER, E, Vampire, Z3, CVC5 — hanno raggiunto livelli di performance straordinari e vengono usati industrialmente per la verifica di hardware (Intel, AMD), compilatori (CompCert), e protocolli crittografici.

PARTE III

I Limiti Strutturali del Connessionismo

Silvano Salvador

5. Ciò che le reti neurali non possono garantire: un'analisi formale

5.1 Il teorema di approssimazione universale: potenza e limiti

Il fondamento teorico del connessionismo è il Teorema di Approssimazione Universale (Cybenko, 1989; Hornik et al., 1989): una rete neurale a singolo strato nascosto con funzione di attivazione sigmoide e larghezza sufficiente è un approssimatore universale di funzioni continue. La versione di Cybenko afferma:

Teorema: Sia $\sigma : \mathbb{R} \to \mathbb{R}$ continua, limitata e non costante. Sia I_n = [0,1]^n. Allora, per ogni $f \in C(I_n)$ e ogni $\varepsilon > 0$, esistono N, w_i, b_i, α_i tali che:

$$| f(x) - \Sigma_{i=1}^{N} \alpha_i \sigma(w_i^T x + b_i) | < \varepsilon \quad \text{per ogni } x \in I_n$$

La dimostrazione si fonda sul teorema di Hahn-Banach: se lo spazio Σ_σ delle combinazioni lineari di traslazioni di σ non fosse denso in C(I_n), esisterebbe per Hahn-Banach un funzionale lineare continuo μ (una misura di Borel) con $\mu(g) = 0$ per ogni $g \in \Sigma_\sigma$ ma $\mu(f) \neq 0$ per qualche f. La condizione $\mu(g) = 0$ implica:

$$\int_{I_n} \sigma(w^T x + b)\, d\mu(x) = 0 \quad \text{per ogni } w \in \mathbb{R}^n, b \in \mathbb{R}$$

Cybenko dimostrò che per funzioni sigmoidi questa condizione forza $\mu = 0$, contraddizione. □

Cosa non afferma questo teorema? Non afferma che la rete trovata dall'addestramento approssima f. Non afferma che la rete generalizza a nuovi dati. Non afferma che la rete ha un comportamento certificabile su tutto il dominio. È un risultato di esistenza, non costruttivo, e richiede

Silvano Salvador

una larghezza in generale non polinomiale in ε e nella dimensione dell'input.

5.2 Generalizzazione fuori distribuzione: il confine della teoria PAC

La teoria PAC-learning (Valiant, 1984) fornisce il quadro formale per comprendere la generalizzazione. Un algoritmo A è (ε, δ)-PAC-learner per una classe di ipotesi $\mathcal{H}$ se, con probabilità almeno $1-\delta$ sull'estrazione casuale di m campioni da una distribuzione D, produce un'ipotesi h con rischio vero:

$$\mathcal{R}(h) \leq \varepsilon$$

dove $\mathcal{R}(h) = P_{(x,y)\sim D}[h(x) \neq y]$. Il Teorema Fondamentale dell'Apprendimento Statistico (Vapnik-Chervonenkis) afferma che una classe $\mathcal{H}$ è PAC-apprendibile se e solo se $VCdim(\mathcal{H}) < \infty$. Il bound di generalizzazione quantitativo è:

$$\mathcal{R}(h) \leq \hat{\mathcal{R}}_m(h) + O\left(\sqrt{(VCdim(\mathcal{H}) \cdot \log(m) + \log(1/\delta)) / m}\right)$$

Il bound vale per nuovi dati estratti dalla stessa distribuzione D che ha generato il training set. Per dati out-of-distribution (OOD) — estratti da una distribuzione $D' \neq D$ — non vi è nessuna garanzia teorica. Un sistema logico, per contro, opera su regole che si applicano uniformemente a tutti gli input che soddisfano le premesse, indipendentemente dalla distribuzione con cui sono stati generati.

Dimostrazione formale dell'assenza di garanzie OOD

Sia $f: \mathbb{R}^n \to \{0,1\}$ la funzione vera, e F_θ la rete addestrata su m campioni da D. Il bound PAC garantisce $\mathcal{R}(F_\theta) \leq \varepsilon$ con probabilità alta. Sia D' una distribuzione con supporto concentrato su regioni

Silvano Salvador

di $\mathbb{R}^n$ a probabilità zero sotto D. Allora il bound PAC dice letteralmente nulla su $\mathcal{R}_{D'}(F_\theta) = P_{(x,y)\sim D'}[F_\theta(x) \neq f(x)]$.

In modo più preciso: il No-Free-Lunch Theorem (Wolpert, 1996) afferma che, in assenza di assunzioni sulla distribuzione dei dati, nessun algoritmo di apprendimento può avere prestazioni attese migliori di un qualsiasi altro su tutti i problemi. La generalizzazione funziona solo perché la distribuzione dei dati reali ha struttura (smoothness, composizionalità, invarianze) che i modelli sfruttano — ma questa struttura non è garantita nelle distribuzioni di deployment.

5.3 La non-certificabilità del comportamento

Per applicazioni safety-critical — controllo del volo, sistemi di guida autonoma, dispositivi medici, software crittografico — non è sufficiente che un sistema si comporti correttamente «quasi sempre»: è necessaria la garanzia formale che esso si comporti correttamente sempre, rispetto a una specifica formalmente definita.

Questa distinzione è fondamentale. Un sistema neurale addestrato a classificare immagini di melanoma con 95% di accuratezza sul test set garantisce che il 5% degli esempi sarà classificato erroneamente — ma non specifica quali esempi, né garantisce il comportamento su immagini non rappresentate nel training set. Per un sistema medico che decide se intervenire chirurgicamente, questa garanzia è insufficiente.

La verifica formale, basata sulla logica, fornisce la struttura per ragionare su tutti gli input simultaneamente: si dimostra che per ogni input x nel dominio di specifica $\mathcal{X}$, l'output F(x) soddisfa la proprietà $\mathcal{P}$(x, F(x)). Questa è una proprietà universalmente quantificata, non statistica.

5.4 Il problema degli adversarial examples

Szegedy et al. (2013) scoprirono un fenomeno sorprendente: le reti neurali profonde sono vulnerabili ad adversarial examples — perturbazioni minuscole dell'input, impercettibili all'occhio umano, che

Silvano Salvador

causano classificazioni errate con alta confidenza. Formalmente, per una rete F_θ e un esempio (x, y), un adversarial example è $x' = x + \delta$ con $\|\delta\| < \varepsilon$ tale che $F_\theta(x') \neq y$.

Il metodo FGSM (Fast Gradient Sign Method, Goodfellow et al., 2014) genera adversarial examples come:

$$x' = x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(F_\theta(x), y))$$

dove $\mathcal{L}$ è la funzione di perdita e $\varepsilon > 0$ è piccolo. La perturbazione è costruita nella direzione del gradiente della perdita rispetto all'input, massimizzando l'errore con il minimo cambiamento di norma.

Gli adversarial examples rivelano che la funzione appresa dalla rete non è geometricamente regolare nel modo in cui le funzioni naturali lo sono: essa ha discontinuità locali che un essere umano non percepisce. Questo è un sintomo strutturale dell'approssimazione statistica: la rete ha appreso una funzione che è statisticamente indistinguibile da f sulla distribuzione dei dati, ma che differisce da f in regioni a bassa probabilità che sono comunque raggiungibili da un avversario.

Robustezza certificata versus robustezza empirica

La robustezza certificata (certified robustness) è la proprietà: per ogni x' con $\|x' - x\| \leq \varepsilon$, $F_\theta(x') = F_\theta(x)$. Questa proprietà può essere verificata formalmente usando strumenti della logica (verifica di reti neurali) e fornisce una garanzia assoluta contro gli adversarial examples nella palla di raggio ε.

La robustezza empirica (empirical robustness) è la proprietà: su un campione di perturbazioni casuali, la rete raramente cambia classificazione. Questa proprietà non fornisce nessuna garanzia contro attacchi avversariali intenzionali.

Nella sicurezza informatica, nella guida autonoma e nella diagnostica medica, solo la robustezza certificata è accettabile. E la robustezza certificata richiede strumenti della logica formale — non statistiche sul test set.

Silvano Salvador

5.5 Il fallimento della composizionalità sistematica

La composizionalità — la capacità di ricombinare concetti appresi in configurazioni nuove — è una proprietà strutturale fondamentale del linguaggio e del ragionamento formale. Sia $\mathcal{L}$ un linguaggio formale generato da una grammatica context-free G. Un sistema apprende $\mathcal{L}$ in modo compositivo se, avendo osservato produzioni di un sottoinsieme finito di G, è in grado di riconoscere e produrre correttamente le produzioni di G intero.

Il benchmark SCAN (Lake e Baroni, 2018) ha mostrato che i modelli sequence-to-sequence, inclusi i Transformer, falliscono sistematicamente nel generalizzare a lunghezze di sequenza o combinazioni di comandi non visti durante l'addestramento. La struttura del fallimento non è graduale ma catastrofica: prestazioni vicine al 100% sui pattern visti durante il training, vicine allo 0% su pattern sistematicamente nuovi.

Un sistema logico non presenta questo problema: le regole di inferenza si applicano a qualsiasi formula ben formata nel linguaggio, indipendentemente dalla sua frequenza di apparizione nel training data. La composizionalità è intrinseca alla struttura sintattica ricorsiva del linguaggio formale.

5.6 Il problema del ragionamento causale

I modelli statistici operano su distribuzioni di probabilità congiunta: apprendono $P(X, Y) = P(Y|X)\ P(X)$ dai dati. Sono pertanto strutturalmente incapaci di distinguere correlazione da causazione. Il framework del do-calculus di Pearl (1995, 2000) formalizza questa distinzione.

Silvano Salvador

L'operatore do: P(Y | do(X = x)) denota la distribuzione di Y quando X è imposto forzatamente al valore x, tagliando tutte le frecce causali entranti in X nel grafo causale. Questa distribuzione è in generale diversa da P(Y | X = x), che è la distribuzione di Y condizionata all'osservazione X = x.

Il Ladder of Causation di Pearl articola tre livelli irriducibili:

- Livello 1 — Associazione: P(Y | X) — «cosa si osserva se X?» — accessibile a qualsiasi modello statistico;
- Livello 2 — Intervento: P(Y | do(X)) — «cosa accade se agisco su X?» — richiede un modello causale strutturale;
- Livello 3 — Controfattuale: P(Y_{X=x} | X=x', Y=y') — «cosa sarebbe accaduto se, contrariamente a ciò che si è osservato, X avesse assunto il valore x?» — richiede equazioni strutturali causali complete.

I sistemi AI che devono ragionare su interventi, su responsabilità, su discriminazione algoritmica, su effetti di politiche: tutti questi richiedono il Livello 2 o Livello 3. Un sistema addestrativo puro — che ha accesso solo a correlazioni statistiche — è confinato al Livello 1 per costruzione. La logica causale di Pearl, nella forma del causal DAG e delle equazioni strutturali, è lo strumento attuale per superare questo confine.

PARTE IV

I Fondamenti Matematici delle Reti Neurali Profonde

6. Architettura e matematica delle reti neurali

6.1 Il neurone formale e il multilayer perceptron

Silvano Salvador

Un singolo neurone artificiale calcola $f(x) = \sigma(w^T x + b)$ dove $x \in \mathbb{R}^n$ è il vettore di input, $w \in \mathbb{R}^n$ è il vettore dei pesi sinaptici, $b \in \mathbb{R}$ è il bias, e $\sigma : \mathbb{R} \to \mathbb{R}$ è la funzione di attivazione. La funzione di attivazione introduce la non-linearità che rende le reti capaci di approssimare funzioni arbitrariamente complesse: senza non-linearità, qualsiasi composizione di trasformazioni lineari rimane lineare.

Le funzioni di attivazione più diffuse nella pratica contemporanea comprendono:

Sigmoide: $\sigma(z) = 1/(1+e^{-z})$, $\sigma'(z) = \sigma(z)(1-\sigma(z))$

Tanh: $\sigma(z) = \tanh(z)$, $\sigma'(z) = 1 - \tanh^2(z)$

ReLU: $\sigma(z) = \max(0,z)$, $\sigma'(z) = \mathbb{1}[z>0]$ (quasi ovunque)

GELU: $\sigma(z) = z \cdot \Phi(z) = z/2 \cdot (1+\mathrm{erf}(z/\sqrt{2}))$

Swish: $\sigma(z) = z/(1+e^{-\beta z})$

Una rete feedforward a L strati è la composizione:

$$F(x) = (f_L \circ f_{L-1} \circ \cdots \circ f_1)(x)$$

dove $f_\ell(h^{(\ell-1)}) = \sigma(W^{(\ell)} h^{(\ell-1)} + b^{(\ell)})$ con $W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ e $b^{(\ell)} \in \mathbb{R}^{n_\ell}$. La preattivazione al livello ℓ è $z^{(\ell)} = W^{(\ell)} h^{(\ell-1)} + b^{(\ell)}$ e l'attivazione è $h^{(\ell)} = \sigma(z^{(\ell)})$.

6.2 Profondità versus larghezza: teoremi di separazione espressiva

I teoremi di separazione espressiva (Telgarsky, 2016) formalizzano il vantaggio computazionale della profondità. Il risultato di Telgarsky è: esistono funzioni computabili da reti ReLU di profondità t e larghezza

Silvano Salvador

$O(1)$ che richiedono, per essere approssimate entro $\varepsilon > 0$ nella norma L_1, reti di profondità 2 e larghezza $\Omega(2^{t/2})$.

La dimostrazione si fonda sull'analisi delle regioni lineari. Una rete ReLU a L strati e larghezza massima n definisce una funzione lineare a tratti, il cui numero di regioni lineari in $\mathbb{R}^{n_0}$ è limitato superiormente da:

$$R(n_0, n_1, \ldots, n_L) \leq \prod_{\ell=1}^{L} \sum_{j=0}^{\min(n_0, n_\ell)} C(n_\ell, j)$$

dove $C(n,j)$ è il coefficiente binomiale. Per $n_\ell \geq n_0$ per ogni ℓ, questo prodotto cresce esponenzialmente in L ma solo polinomialmente nella larghezza per $L = 1$. Ne segue che la profondità conferisce un vantaggio espressivo esponenziale per funzioni oscillanti (ad esempio, funzioni triangolari iterate).

6.3 L'algoritmo di retropropagazione: derivazione completa

L'addestramento di una rete neurale è un problema di ottimizzazione non convessa. Dati un training set $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ e una funzione di perdita $\mathcal{L}$, si minimizza la perdita empirica:

$$\hat{\mathcal{L}}(\theta) = (1/N) \sum_{i=1}^{N} \mathcal{L}(F_\theta(x^{(i)}), y^{(i)})$$

La funzione di perdita standard per la classificazione multiclasse è la cross-entropy:

$$\mathcal{L}_{CE}(\hat{y}, y) = -\sum y \log(\hat{y})$$

dove $\hat{y} = \text{softmax}(z^{(L)})$ e y è il vettore one-hot. Minimizzare la cross-entropy equivale a minimizzare la divergenza KL tra distribuzione empirica e distribuzione modello, poiché $D_{KL}(P_{data} \| P_\theta) = H(P_{data}, P_\theta) - H(P_{data})$ e $H(P_{data})$ è costante rispetto a θ.

Silvano Salvador

L'algoritmo di retropropagazione calcola efficientemente i gradienti. Si definiscono i gradienti di errore $\delta^{(\ell)} = \partial \mathcal{L}/\partial z^{(\ell)}$. La fase backward è:

$$\delta^{(L)} = \hat{y} - y \qquad \text{[per softmax + cross-entropy]}$$

$$\delta^{(\ell)} = (W^{(\ell+1)})^T \delta^{(\ell+1)} \odot \sigma'(z^{(\ell)}) \quad \text{[regola della catena]}$$

I gradienti rispetto ai parametri sono:

$$\partial \mathcal{L}/\partial W^{(\ell)} = \delta^{(\ell)} (h^{(\ell-1)})^T$$

$$\partial \mathcal{L}/\partial b^{(\ell)} = \delta^{(\ell)}$$

La derivazione della regola backward segue direttamente dalla regola della catena multivariata: $\partial \mathcal{L}/\partial z_i^{(\ell)} = \Sigma \; (\partial \mathcal{L}/\partial z_j^{(\ell+1)}) \cdot W_{ji}^{(\ell+1)} \cdot \sigma'(z_i^{(\ell)})$, che in forma vettoriale diventa la formula sopra. □

6.4 Il problema del gradiente che svanisce e le soluzioni architetturali

Il problema del gradiente che svanisce (Hochreiter, 1991; Bengio et al., 1994) emerge dalla struttura prodotto della retropropagazione. Il gradiente al primo strato è:

$$\delta^{(1)} = J_L \cdot J_{L-1} \cdot \cdots \cdot J_2 \cdot \delta^{(L)}$$

dove $J_\ell = \mathrm{diag}(\sigma'(z^{(\ell)})) \cdot W^{(\ell+1)}$ è la matrice jacobiana dello strato ℓ. Se $\|J_\ell\|_2 < 1$ per ogni strato, il prodotto tende a zero esponenzialmente in L. Per la sigmoide, $\sigma'(z) \leq 1/4$, rendendo inevitabile la scomparsa del gradiente per reti molto profonde.

Silvano Salvador

La soluzione architetturale più elegante è quella delle reti residuali (ResNets, He et al., 2016):

$$h^{(\ell+1)} = h^{(\ell)} + R(h^{(\ell)}, \theta_\ell)$$

Il gradiente per una rete residuale è:

$$\partial \mathcal{L}/\partial h^{(\ell)} = \partial \mathcal{L}/\partial h^{(L)} \cdot \prod_{k=\ell}^{L-1} (I + \partial R_k/\partial h^{(k)})$$

Il termine identità I garantisce che il gradiente si propaghi, almeno parzialmente, ininterrotto fino agli strati più profondi. Le Neural ODE (Chen et al., 2018) interpretano le ResNet come discretizzazioni di equazioni differenziali ordinarie $dh/dt = R(h(t), \theta(t))$, e usano il metodo aggiunto per calcolare i gradienti in costo di memoria O(1).

7. Il meccanismo di attenzione e i Transformer

7.1 Il problema delle dipendenze a lungo raggio

Nelle reti ricorrenti (LSTM, GRU), l'informazione si propaga sequenzialmente attraverso gli stati nascosti, e la distanza tra due token nella sequenza si traduce in un cammino computazionale proporzionalmente lungo, soggetto al problema del gradiente che svanisce. Il meccanismo di attenzione risolve questo problema in modo elegante: calcola, per ogni coppia di posizioni nella sequenza, un peso che esprime la rilevanza reciproca, permettendo il flusso diretto di informazione tra posizioni arbitrariamente distanti.

7.2 Scaled dot-product attention: derivazione e giustificazione

Date matrici di query $Q \in \mathbb{R}^{n \times d_k}$, chiavi $K \in \mathbb{R}^{m \times d_k}$ e valori $V \in \mathbb{R}^{m \times d_v}$, la Scaled Dot-Product Attention è:

Silvano Salvador

$$\text{Attn}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

Il prodotto $QK^T \in \mathbb{R}^{n \times m}$ è la matrice dei punteggi di attenzione: l'elemento (i,j) misura la compatibilità tra la i-esima query e la j-esima chiave. La scalatura per $\sqrt{d_k}$ ha una giustificazione statistica precisa: se le componenti di Q e K sono variabili casuali indipendenti con media 0 e varianza 1, il prodotto q^Tk ha varianza d_k. La scalatura per $\sqrt{d_k}$ ripristina la varianza unitaria, mantenendo i gradienti della softmax in un regime non saturo. Per d_k grande senza scalatura, la softmax si saturava verso distribuzioni quasi one-hot, con gradienti vanishing.

7.3 Multi-head attention e architettura completa

La Multi-Head Attention applica h meccanismi di attenzione paralleli:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h) \cdot W^O$$

$$\text{head}_i = \text{Attn}(QW_i^Q, KW_i^K, VW_i^V)$$

con proiezioni apprendibili $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. Tipicamente $d_k = d_v = d_{model}/h$.

Un blocco Transformer completo combina Multi-Head Attention con reti feedforward position-wise e Layer Normalization con connessioni residuali:

$$h'^{(\ell)} = \text{LayerNorm}(h^{(\ell-1)} + \text{MultiHead}(h^{(\ell-1)}, h^{(\ell-1)}, h^{(\ell-1)}))$$

$$h^{(\ell)} = \text{LayerNorm}(h'^{(\ell)} + \text{FFN}(h'^{(\ell)}))$$

dove $\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$ con dimensione interna $4d_{model}$. La Layer Normalization normalizza per ogni token lungo la dimensione del modello, differentemente dalla Batch Normalization che normalizza lungo il batch.

Silvano Salvador

7.4 Encoding posizionale e rotary embeddings

Il meccanismo di attenzione è invariante alla permutazione: senza informazione posizionale, il Transformer tratterebbe la sequenza come un insieme non ordinato. La soluzione originale (Vaswani et al., 2017) usa encoding sinusoidali fissi:

$$PE(pos, 2i) = \sin(pos / 10000^{2i/d_{model}})$$

$$PE(pos, 2i+1) = \cos(pos / 10000^{2i/d_{model}})$$

La proprietà chiave è che PE(pos + k) è una trasformazione lineare di PE(pos), facilitando l'apprendimento della posizione relativa. Le RoPE (Rotary Position Embeddings, Su et al., 2021) codificano la posizione relativa direttamente nel prodotto query-chiave:

$$q^T k \rightarrow (R_{pos_q}\, q)^T (R_{pos_k}\, k) = q^T R_{pos_k - pos_q}\, k$$

dove R_θ è una matrice di rotazione nel piano complesso. RoPE è ora lo standard nei modelli di linguaggio di grande scala (LLaMA, Mistral, Claude) perché consente una migliore generalizzazione a lunghezze di sequenza superiori a quelle di addestramento.

7.5 Ottimizzatori adattativi: Adam e varianti

La discesa del gradiente stocastica nella sua forma base aggiorna tutti i parametri con lo stesso tasso di apprendimento. L'ottimizzatore Adam (Kingma e Ba, 2015) mantiene stime del primo e secondo momento dei gradienti:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\, g_t \qquad \text{[media mobile esponenziale]}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2 \qquad \text{[varianza mobile esponenziale]}$$

Le stime corrette per il bias di inizializzazione sono $\hat{m}_t = m_t/(1-\beta_1^t)$ e $\hat{v}_t = v_t/(1-\beta_2^t)$. L'aggiornamento è:

Silvano Salvador

$$\theta_t = \theta_{t-1} - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$$

con valori tipici $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$. AdamW (Loshchilov e Hutter, 2019) separa il weight decay dall'aggiornamento adattivo per ripristinare la corretta interpretazione bayesiana della regolarizzazione L_2, evitando l'interazione anomala con il denominatore adattivo.

8. Teoria statistica dell'apprendimento: i fondamenti formali della generalizzazione

8.1 La dimensione VC e il Teorema Fondamentale

La dimensione di Vapnik-Chervonenkis (VC dimension) di una classe di ipotesi $\mathcal{H}$ è definita come la cardinalità del più grande insieme di punti che $\mathcal{H}$ può shattering — classificare in tutti i 2^m modi possibili:

$$\text{VCdim}(\mathcal{H}) = \sup\{m \in \mathbb{N} : \exists\, S \subseteq \mathcal{X},\ |S| = m,\ \Pi_{\mathcal{H}}(S) = 2^m\}$$

dove $\Pi_{\mathcal{H}}(S) = |\{(h(x_1), \ldots, h(x_m)) : h \in \mathcal{H}\}|$. Il Lemma di Sauer-Shelah afferma: se $\text{VCdim}(\mathcal{H}) = d$, allora $\Pi_{\mathcal{H}}(S) \leq \sum_{i=0}^{d} C(|S|, i)$, che cresce polinomialmente in $|S|$ per d finito.

Teorema Fondamentale (Vapnik-Chervonenkis): Una classe di ipotesi binarie $\mathcal{H}$ è PAC-apprendibile se e solo se $\text{VCdim}(\mathcal{H}) < \infty$. Il campione di complessità (sample complexity) è:

$$m_{\mathcal{H}}(\varepsilon, \delta) = O((\text{VCdim}(\mathcal{H}) + \log(1/\delta)) / \varepsilon^2)$$

8.2 Il PAC-Bayes bound e la connessione con l'inferenza bayesiana

Silvano Salvador

Teorema PAC-Bayes (McAllester, 1999): Per ogni distribuzione prior P sui parametri, per ogni distribuzione posterior Q, con probabilità almeno $1-\delta$:

$$\mathcal{R}(Q) \leq \hat{\mathcal{R}}_m(Q) + \sqrt{((D_KL(Q \| P) + \log(m/\delta)) / (2m))}$$

dove $\mathcal{R}(Q) = E_\{h\sim Q\}[\mathcal{R}(h)]$ è il rischio vero rispetto a Q e $\hat{\mathcal{R}}_m(Q)$ è il rischio empirico. Il bound generalizza il bound VC sostituendo la complessità combinatoria con la divergenza KL tra prior e posterior — una misura informazionale della complessità effettiva del modello.

Questo bound ha un'interpretazione elegante in termini del Principio di Lunghezza di Descrizione Minima (MDL, Rissanen, 1978): il modello migliore per un insieme di dati è quello che consente la loro descrizione più compatta. Il costo di codifica totale è:

$$L_total(M, D) = L(M) + L(D \mid M) \leftrightarrow -\log P(M) - \log P(D|M) \propto -\log P(M|D)$$

Minimizzare L_total equivale a massimizzare la probabilità a posteriori — l'MDL è la formulazione informazionale dell'inferenza bayesiana.

8.3 Scaling laws e capacità emergenti

Le leggi di scala (Kaplan et al., 2020) descrivono come le prestazioni dei grandi modelli linguistici crescono con la scala in modo regolare:

$$\mathcal{L}(N) \approx (N_c / N)^\{\alpha_N\} \quad \text{con } \alpha_N \approx 0.076$$

$$\mathcal{L}(D) \approx (D_c / D)^\{\alpha_D\} \quad \text{con } \alpha_D \approx 0.095$$

$$\mathcal{L}(C) \approx (C_c / C)^\{\alpha_C\} \quad \text{con } \alpha_C \approx 0.050$$

dove $\mathcal{L}$ è la perdita di cross-entropy, N è il numero di parametri, D la dimensione del dataset e C la computazione. Queste leggi di potenza prevedono una crescita continua e regolare delle prestazioni —

Silvano Salvador

incompatibile con le capacità emergenti discrete osservate empiricamente (Wei et al., 2022).

Le capacità emergenti sono abilità che appaiono improvvisamente al superamento di soglie dimensionali, senza essere state esplicitamente ottimizzate: aritmetica multi-cifra, soluzione di analogie, ragionamento multi-step, traduzione zero-shot. La loro imprevedibilità ha implicazioni dirette per la sicurezza: la valutazione del comportamento di un modello più grande non può essere inferita con certezza dall'analisi di modelli più piccoli.

PARTE V

La Logica come Strumento di Certificazione e Verifica

9. Verifica formale di reti neurali: fondamenti e metodi

9.1 Il problema della verifica neurale e la sua difficoltà computazionale

La verifica formale di reti neurali affronta il problema di dimostrare proprietà di sicurezza di reti addestrate usando strumenti della logica matematica e dell'analisi formale. Formalmente, data una rete $F_\theta : \mathbb{R}^n \to \mathbb{R}^m$, una regione di input $\mathscr{X} \subseteq \mathbb{R}^n$ e una specifica di output $\mathscr{P} \subseteq \mathbb{R}^m$, si vuole verificare:

$$\forall x \in \mathscr{X}: F_\theta(x) \in \mathscr{P}$$

Silvano Salvador

Questa è una proprietà universalmente quantificata su un insieme continuo — non una statistica su un campione finito. Per reti ReLU, la difficoltà computazionale è precisamente caratterizzata:

Teorema (Katz et al., 2017): Determinare se una rete neurale feedforward a strati nascosti con attivazione ReLU soddisfa una specifica lineare su un insieme di input linearmete vincolato è un problema NP-completo.

La dimostrazione è per riduzione da SAT. L'attivazione ReLU $\sigma(z) = \max(0, z)$ introduce una biforcazione: ogni neurone è attivo ($z > 0$, $\sigma(z) = z$) o inattivo ($z \leq 0$, $\sigma(z) = 0$). Per n neuroni ReLU, ci sono 2^n possibili combinazioni di stati, e determinare quale si realizza per un dato input è equivalente a risolvere un problema di programmazione lineare intera.

9.2 Reluplex e Marabou: verifica mediante SMT solving

Il sistema Reluplex (Katz et al., 2017) estende il metodo Simplex per la programmazione lineare per gestire l'attivazione ReLU come un vincolo non lineare. L'algoritmo mantiene un tableau simplex aumentato con variabili ausiliarie per ogni neurone ReLU: $f_i = \text{ReLU}(b_i)$ dove b_i è la preattivazione e f_i l'uscita.

La procedura applica iterativamente:

1. Simplex Phase: si risolve il problema di programmazione lineare ignorando i vincoli ReLU non soddisfatti;
2. Fixing Phase: si identificano i neuroni in cui il vincolo ReLU è violato ($f_i \neq \text{ReLU}(b_i)$) e si aggiornano i valori;
3. Case-splitting: se il fixing fallisce, si divide sul caso $b_i \leq 0$ (neurone inattivo) oppure $b_i \geq 0$ (neurone attivo).

Silvano Salvador

Il successore Marabou (Katz et al., 2019) incorpora tecniche di Theory Propagation che permettono di derivare implicazioni logiche dalle attivazioni senza eseguire case-splitting esplicito, riducendo significativamente la dimensione dell'albero di ricerca per reti tipiche.

Applicazione pratica: ACAS Xu

Reluplex è stato originariamente sviluppato per verificare proprietà di sicurezza di ACAS Xu (Airborne Collision Avoidance System X for Unmanned Aircraft), una rete neurale a 6 strati nascosti con 300 neuroni per strato usata per il controllo del traffico aereo.

Le proprietà verificate includevano: se l'aeromobile invasore si trova alla sinistra dell'aeromobile protetto e si muove verso destra, il sistema raccomanda virata a sinistra. Questa è esattamente la forma $\forall x \in \mathscr{X}$: $F_0(x) \in \mathscr{P}$ con $\mathscr{X}$ e $\mathscr{P}$ descritti da disuguaglianze lineari.

Reluplex ha rilevato violazioni in alcune delle 10 reti ACAS Xu verificate, dimostrando che la verifica formale fornisce garanzie che il testing su campioni non può fornire.

9.3 Abstract interpretation: metodi sound per la verifica scalabile

Il metodo delle Abstract Interpretations (Gehr et al., 2018 — DeepZ, DeepPoly) approssima la propagazione di insiemi di input attraverso la rete usando domini astratti — poliedri convessi che possono essere propagati efficientemente attraverso ogni strato.

Per un input $x \in \mathscr{X} = \{x : Ax \leq b\}$ (poliedro), la propagazione attraverso uno strato lineare $W x + c$ produce un altro poliedro. La difficoltà è la propagazione attraverso l'attivazione ReLU, che è non lineare. DeepPoly utilizza approssimazioni lineari: per ogni neurone con preattivazione z nell'intervallo [l, u], si usa:

- Se $u \leq 0$: neurone inattivo, output = 0;
- Se $l \geq 0$: neurone attivo, output = z;

Silvano Salvador

- Se $l < 0 < u$: approssimazione lineare: output $\leq u(z-l)/(u-l)$ (vincolo superiore); output ≥ 0 o output $\geq z$ (vincolo inferiore, scelta che minimizza l'area del triangolo).

Il risultato è un poliedro astratto che sovra-approssima l'insieme raggiungibile: ogni punto raggiunto dalla rete per input in $\mathcal{X}$ appartiene al poliedro astratto, ma non viceversa. La verifica è sound: se il poliedro astratto è contenuto in $\mathcal{P}$, la proprietà è verificata per ogni input in $\mathcal{X}$. La verifica può fallire per via della sovra-approssimazione (falsi negativi), ma non può affermare falsamente che una proprietà è verificata quando non lo è.

9.4 Robustezza certificata: dalle garanzie formali alla pratica

La robustezza certificata è la proprietà: per ogni x' con $\|x' - x\|_p \leq \varepsilon$, la classificazione della rete non cambia. Le tecniche per ottenere robustezza certificata durante l'addestramento includono:

- Interval Bound Propagation (IBP): si propagano intervalli attraverso la rete durante il training e si minimizza una perdita robusta che include il termine di verifica;
- CROWN e α-CROWN: si usano bound lineari parametrici più stretti, rilassati, che forniscono approssimazioni più precise degli intervalli;
- Randomized Smoothing (Cohen et al., 2019): si addestra un classificatore di base g e si definisce il classificatore liscio $f(x) = \text{argmax}_c \, P(g(x+\varepsilon) = c)$ per $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Si può certificare che $f(x) = f(x')$ per tutti x' con $\|x' - x\|_2 \leq \sigma \cdot \Phi^{-1}(p_A)$, dove p_A è la probabilità della classe più probabile sotto rumore gaussiano.

Il Randomized Smoothing è importante perché fornisce garanzie di robustezza certificata scalabili a reti molto grandi (ImageNet scale), a differenza dei metodi esatti che sono computazionalmente proibitivi per tali scale.

Silvano Salvador

10. Logica temporale e model checking per sistemi AI

10.1 Logiche per la specifica di sistemi dinamici

La Logica Temporale Lineare (LTL) e la Computation Tree Logic (CTL) forniscono linguaggi formali per esprimere proprietà di sistemi dinamici su sequenze di stati. In LTL, le formule sono costruite mediante gli operatori:

- □φ — 'φ è vero in tutti gli stati futuri' (globally/always);
- ◇φ — 'φ è vero in almeno uno stato futuro' (eventually);
- Xφ — 'φ è vero nello stato successivo' (next);
- φ U ψ — 'φ è vero fino a che ψ diventa vero' (until).

Le proprietà fondamentali dei sistemi AI possono essere espresse in LTL:

Safety: □(¬Collisione) [il veicolo non collide mai]

Liveness: □(Richiesta → ◇Risposta) [ogni richiesta riceve risposta]

Fairness: □◇Serving_A ∧ □◇Serving_B [ogni agente è servito infinitamente spesso]

Progress: □(¬Terminato → ◇Progress) [il sistema progredisce sempre]

In CTL, le formule quantificano sui cammini dell'albero di computazione:

- AGφ — 'per tutti i cammini, sempre φ' (AG = □ universale);
- EFφ — 'esiste un cammino in cui eventualmente φ' (EF = ◇ esistenziale);

Silvano Salvador

- A[φ U ψ] — 'per tutti i cammini, φ until ψ'.

10.2 Model checking: algoritmi e complessità

Il model checking per CTL su sistemi di transizione finiti è P-completo (Clarke et al., 1986). L'algoritmo fondamentale calcola il più grande punto fisso dell'operatore di predecessori. Per una formula φ e un sistema di transizione $M = \langle S, \rightarrow, L \rangle$ con insiemi di stati S, relazione di transizione → e funzione di labeling L:

La soddisfacibilità di CTL su un sistema M si verifica mediante un algoritmo di visita del grafo che calcola ricorsivamente l'insieme degli stati che soddisfano ogni sottoformula. La complessità è $O(|M| \cdot |\varphi|)$ dove $|M|$ è la dimensione del sistema e $|\varphi|$ la lunghezza della formula.

Il model checking per LTL è PSPACE-completo: la costruzione dell'automa di Büchi per una formula LTL di lunghezza n richiede un automa con 2^O(n) stati, e il problema di vuotezza dell'intersezione tra l'automa del sistema e l'automa della negazione della formula è PSPACE-completo.

Strumenti industriali di model checking

SPIN (Holzmann, 1987): verificatore open-source per protocolli distribuiti, usa LTL su automi di Promela;

nuSMV (Clarke et al.): verificatore simbolico per CTL e LTL, usa Binary Decision Diagrams (BDD) per rappresentare l'insieme degli stati in modo compatto;

TLA+ (Lamport): linguaggio di specifica formale usato da Amazon AWS, Microsoft Azure, Intel per verificare protocolli di sistemi distribuiti; le violazioni rilevate nella specifica di sistemi reali hanno giustificato l'adozione industriale.

Silvano Salvador

10.3 Estensione al reinforcement learning: logica per la specifica della ricompensa

L'apprendimento per rinforzo (RL) è il paradigma in cui un agente apprende una politica massimizzando il ritorno atteso cumulativo. Formalmente si usa il framework dei Markov Decision Process (MDP): una quintupla $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ dove la politica ottima π^* soddisfa le equazioni di Bellman:

$$V^*(s) = \max_a [R(s,a) + \gamma \Sigma_{s'} T(s,a,s') V^*(s')]$$

La connessione con la logica formale emerge in due direzioni critiche. Prima direzione: la specifica della funzione di ricompensa R(s,a). Definire R in modo che catturi accuratamente gli obiettivi desiderati — senza reward hacking, senza effetti collaterali indesiderati, con garanzie di sicurezza — è un problema di logica formale applicata. Il reward shaping scorretto è una delle fonti principali di comportamenti indesiderati nei sistemi RL.

La Temporal Logic Mission Specification (TLTM) consente di definire obiettivi di RL come formule LTL: la ricompensa è progettata automaticamente a partire dalla specifica formale, garantendo che il comportamento ottimo rispetto alla ricompensa coincida con il comportamento desiderato rispetto alla specifica. Questo approccio elimina la necessità di definire manualmente la funzione di ricompensa e fornisce garanzie formali sul comportamento dell'agente addestrato.

Seconda direzione: la verifica di politiche apprese. Data una politica neurale π e una proprietà LTL φ, si vuole verificare che π soddisfi φ. Per MDP finiti questo è decidibile; per spazi di stati continui con politiche neurali, si applicano tecniche di verifica probabilistica basata su campionamento (Statistical Model Checking, Legay e Delahaye, 2011).

Silvano Salvador

11. Sistemi di tipi dipendenti e la matematica formale certificata

11.1 La corrispondenza di Curry-Howard-Lambek

Uno dei risultati più profondi della logica matematica applicata all'informatica è la corrispondenza di Curry-Howard (1934/1969), generalizzata da Lambek (1969) alla teoria delle categorie. Essa stabilisce un isomorfismo trilaterale:

LOGICAProposizioneDimostrazioneImplicazione A → BCongiunzione A ∧ BDisgiunzione A ∨ BFalso ⊥Normalizzazione della dimostrazione	TEORIA DEI TIPITipoProgramma di quel tipoTipo funzione A → BTipo prodotto A × BTipo somma A + BTipo vuoto 0Esecuzione del programma

Formalmente, nel lambda-calcolo semplicemente tipato, la regola di tipaggio Γ ⊢ M : A corrisponde alla regola di inferenza Γ ⊢_logica A con prova M, dove Γ è il contesto (ipotesi/variabili tipate), M è il termine (programma/dimostrazione) e A è il tipo (proposizione). Le regole di introduzione ed eliminazione per i connettivi logici corrispondono esattamente alle regole per i costruttori di tipo.

11.2 Tipi dipendenti: quando i tipi dipendono dai valori

Nei sistemi di tipi semplici, i tipi non possono dipendere dai valori. Nei sistemi di tipi dipendenti (Martin-Löf, 1975), i tipi possono dipendere dai termini. Questo consente di esprimere proprietà matematiche come tipi:

Vect(α, n) = tipo dei vettori di tipo α e lunghezza n

Silvano Salvador

Even(n) = tipo delle dimostrazioni che n è pari

Prime(n) = tipo delle dimostrazioni che n è primo

Una funzione f : (n : ℕ) → Vect(α, n) → Vect(α, n+1) ha tipo che garantisce che il vettore di output ha sempre esattamente un elemento in più dell'input — per costruzione, senza possibilità di errore a runtime. Questo è il principio della Correct by Construction programming: le proprietà desiderate sono codificate nel tipo, e il verificatore di tipi garantisce la loro soddisfazione.

I proof assistant basati su tipi dipendenti — Coq (ora Rocq), Agda, Lean 4, Isabelle/HOL — permettono di formalizzare definizioni matematiche, enunciare teoremi come tipi e costruire prove come programmi. Un programma Coq che compila è, per costruzione, matematicamente corretto rispetto alla sua specifica.

11.3 Formalizzazioni matematiche monumentali

La potenza dei proof assistant è testimoniata da alcune formalizzazioni matematiche di portata straordinaria:

- Teorema di Feit-Thompson (Gonthier et al., 2013): ogni gruppo di ordine dispari è risolubile. La formalizzazione in Coq ha richiesto 6 anni di lavoro, 170.000 righe di codice, e ha identificato alcuni errori minori nella dimostrazione a matita originale di 255 pagine.
- Congettura di Kepler / Teorema di Hales (progetto Flyspeck, 2017): la disposizione di sfere in spazio euclideo tridimensionale più densa è quella della pila di cannonballe (face-centered cubic). La dimostrazione originale di Hales (1998) dipendeva da calcoli

Silvano Salvador

computerizzati che i referee non potevano verificare; il progetto Flyspeck li ha formalizzati completamente in HOL Light e Isabelle.

- Kernel seL4 (Klein et al., 2009): il kernel di un sistema operativo di 8,700 righe di C è stato formalmente verificato in Isabelle/HOL. La specifica formale garantisce proprietà di integrità della memoria, isolamento dei processi e correttezza della gestione delle syscall — proprietà impossibili da verificare mediante testing esaustivo.
- Compilatore CompCert (Leroy, 2009): un compilatore C formalmente verificato in Coq che garantisce che il codice assembly generato abbia la stessa semantica del codice C sorgente. Usato in applicazioni avioniche (Airbus).

11.4 LLM come proof-search: la collaborazione neuro-simbolica nelle dimostrazioni

La combinazione di modelli linguistici con proof assistant rappresenta la forma più matura di architettura neuro-simbolica nel dominio del ragionamento matematico. L'idea fondamentale è la separazione dei ruoli: il modello neurale suggerisce tattiche di dimostrazione in modo probabilistico; il proof assistant verifica ogni passo in modo formale.

Lean Copilot (Han et al., 2024) integra un modello linguistico nel loop di interazione con Lean 4: il matematico enuncia il teorema, il modello suggerisce la prossima tattica o la dimostrazione completa, Lean verifica. Il modello può essere interrogato per completamento automatico di tattiche (come GitHub Copilot per il codice), e la verifica di Lean garantisce che nessuna tattica errata venga accettata.

Silvano Salvador

AlphaProof (DeepMind, 2024) applica reinforcement learning nel loop del proof assistant: il modello impara a suggerire tattiche che vengono accettate da Lean, ricevendo ricompensa positiva per ogni passo di dimostrazione valido e ricompensa finale per la dimostrazione completa. Questo ha prodotto il primo sistema AI in grado di risolvere problemi della competizione IMO a livello di medaglia d'argento.

L'architettura fondamentale

In tutti questi sistemi, la struttura è invariante: il componente neurale propone, il componente simbolico dispone. La garanzia di correttezza risiede interamente nel verificatore formale. Rimuovere il componente simbolico significherebbe eliminare l'unica garanzia di correttezza: il modello neurale, da solo, produce dimostrazioni che sembrano plausibili ma contengono errori sottili.

Questo non è un dettaglio ingegneristico: è la struttura concettuale che risponde definitivamente alla domanda 'la logica ha ancora senso?'. Nell'era dei sistemi AI più potenti mai costruiti, la logica è l'arbitro finale della correttezza. Il deep learning ha ampliato lo spazio di ricerca delle dimostrazioni; la logica è rimasta l'unico giudice affidabile di cosa sia una dimostrazione.

PARTE VI

Logica Probabilistica, Teoria dell'Informazione e Architetture Neuro-Simboliche

12. La teoria dell'informazione come fondamento trasversale

12.1 Shannon e la misura matematica dell'informazione

Silvano Salvador

Uno degli edifici teorici che attraversa trasversalmente sia il paradigma simbolico che quello connessionista, fungendo da linguaggio comune e strumento di misura condiviso, è la teoria dell'informazione fondata da Claude Elwood Shannon nel 1948. La nozione fondamentale è quella di entropia di Shannon di una variabile aleatoria discreta X con distribuzione $P(X = x_i) = p_i$:

$$H(X) = -\Sigma_i \, p_i \log_2 p_i$$

misurata in bit. L'entropia misura l'incertezza media associata alla variabile X, ovvero il numero medio di bit necessari per codificare ottimalmente i suoi valori. Il Teorema di Shannon (1948) afferma che il codice ottimale per una sorgente discreta richiede H(X) bit per simbolo in media, e che nessun codice può fare meglio (Teorema di codifica della sorgente).

Le proprietà fondamentali dell'entropia sono:

- $H(X) \geq 0$, con uguaglianza sse P è una distribuzione di Dirac (certezza completa);
- $H(X) \leq \log_2 |\mathcal{X}|$, con uguaglianza sse P è uniforme su $\mathcal{X}$ (massima incertezza);
- Per variabili indipendenti: $H(X,Y) = H(X) + H(Y)$ (additività);
- In generale: $H(X,Y) \leq H(X) + H(Y)$ (subadditività);
- $H(X|Y) \leq H(X)$ — il condizionamento riduce l'entropia.

L'informazione mutua $I(X;Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y)$ misura la riduzione dell'incertezza su X fornita dalla conoscenza di Y e costituisce la misura naturale della dipendenza statistica tra due variabili, generalizzando la correlazione lineare a dipendenze arbitrarie.

Silvano Salvador

12.2 La divergenza di Kullback-Leibler e la geometria dell'informazione

La divergenza di Kullback-Leibler tra due distribuzioni è:

$$D_{KL}(P \| Q) = \Sigma_x P(x) \log(P(x)/Q(x))$$

La KL divergence misura quanto Q è un «cattivo modello» per dati generati da P. Essa soddisfa: $D_{KL}(P \| Q) \geq 0$, con uguaglianza sse P = Q q.o. (disuguaglianza di Jensen); $D_{KL}(P \| Q) \neq D_{KL}(Q \| P)$ in generale (non è una metrica). La divergenza Jensen-Shannon $J(P,Q) = (D_{KL}(P\|Q) + D_{KL}(Q\|P))/2$ è simmetrica e limitata in [0,1].

L'addestramento di modelli probabilistici corrisponde, nella maggior parte dei casi, alla minimizzazione della KL divergenza tra distribuzione empirica P_{data} e distribuzione modello P_θ:

$$\min_\theta D_{KL}(P_{data} \| P_\theta) = \min_\theta H(P_{data}, P_\theta) - H(P_{data}) = \max_\theta \Sigma_i \log P_\theta(x^{(i)})$$

stabilendo la connessione fondamentale tra teoria dell'informazione e massima verosimiglianza.

La geometria dell'informazione (Amari, 1985) studia lo spazio delle distribuzioni di probabilità come varietà differenziale dotata della metrica di Fisher:

$$g_{ij}(\theta) = E_{P_\theta}[\partial_i \log P_\theta(X) \cdot \partial_j \log P_\theta(X)]$$

Questa metrica è la geometria naturale delle famiglie esponenziali e fonda la Natural Gradient Descent: $\theta_{t+1} = \theta_t - \eta \cdot F(\theta_t)^{-1} \nabla_\theta \mathcal{L}(\theta_t)$, che è invariante alle reparametrizzazioni della distribuzione.

12.3 Connessioni tra attenzione e inferenza bayesiana

Silvano Salvador

Vi è una connessione formale, raramente discussa nel dettaglio, tra il meccanismo di attenzione dei Transformer e la struttura dell'inferenza bayesiana. La matrice di attenzione A = softmax(QK^{T} / √d_k) ∈ ℝ^{n × n} è stocastica per righe: ogni riga somma a 1 e ha componenti non negative. L'elemento A_{ij} può essere interpretato come la probabilità condizionale che, nella produzione dell'output alla posizione i, si «attinga» al valore alla posizione j.

Nel formalismo delle reti bayesiane, un meccanismo analogo appare nelle mixture models e nelle Gaussian Process regression: il valore atteso della variabile target dato l'input è una media ponderata dei valori di training, con pesi determinati dalla similarità (kernel) tra il punto di query e i punti di training — esattamente la struttura dell'attenzione con kernel esponenziale. Questa connessione è resa esplicita nei Gaussian Process Transformers (Nguyen e Salazar, 2019).

13. Logica probabilistica e ragionamento in condizioni di incertezza

13.1 Dalla logica classica alla logica probabilistica

La logica classica opera con valori di verità binari — vero o falso — e il principio del terzo escluso garantisce che ogni proposizione ben formata sia o vera o falsa. Questa struttura è perfettamente adeguata al ragionamento matematico puro. Essa è tuttavia inadeguata al ragionamento in condizioni di incertezza pratica.

Nilsson (1986) introdusse la probabilistic logic come estensione della logica del primo ordine: a ogni formula viene assegnata una probabilità $P(\varphi) = p \in [0,1]$. Un database logico-probabilistico è un insieme di

Silvano Salvador

coppie (φ_i, p_i) e il problema dell'inferenza consiste nel calcolare i valori ammissibili di P(ψ) per una query ψ, data la struttura di probabilità.

L'inferenza bayesiana in reti bayesiane — grafi orientati aciclici (DAG) con la fattorizzazione di Markov $P(V_1,...,V) = \prod_i P(V_i \mid Pa(V_i))$ — è NP-difficile in generale (Cooper, 1990). L'algoritmo belief propagation di Pearl (1988) risolve esattamente l'inferenza su reti con struttura ad albero, con complessità O(n · d^k) dove d è la cardinalità massima dei domini e k la dimensione massima degli insiemi di genitori.

13.2 Markov Logic Networks: la sintesi più ambiziosa

Le Markov Logic Networks (MLN, Richardson e Domingos, 2006) costituiscono il tentativo più sistematico di integrare logica del primo ordine e modelli probabilistici. Una MLN è un insieme di coppie (F_i, w_i) dove F_i è una formula FOL e $w_i \in \mathbb{R}$ è un peso. Data una base di costanti $\mathscr{C} = \{c_1, ..., c\}$, la MLN definisce una distribuzione di Gibbs:

$$P(\omega) = (1/Z) \exp(\Sigma_i w_i n_i(\omega))$$

dove $n_i(\omega)$ è il numero di grounding della formula F_i che risultano veri nell'assegnazione ω, e $Z = \Sigma_{\{\omega'\}} \exp(\Sigma_i w_i n_i(\omega'))$ è la funzione di partizione. Il peso w_i codifica la «forza» della formula come vincolo morbido:

- $w_i \to +\infty$: la formula F_i diventa un vincolo rigido, si approssima al modello logico classico;
- $w_i = 0$: la formula F_i è irrilevante per la distribuzione;
- $w_i < 0$: la formula F_i è penalizzata, rendendo improbabili le assegnazioni che la soddisfano.

L'inferenza in MLN (calcolare P(V_query | V_evidence = v_e)) è computazionalmente più difficile che nelle reti bayesiane: la funzione di

Silvano Salvador

partizione Z è #P-difficile da calcolare. Le approssimazioni pratiche usano MC-SAT (Poon e Domingos, 2006) — una variante di MCMC che campiona da una distribuzione proposizionale — o tecniche di inferenza variazionale.

Applicazioni delle Markov Logic Networks

Entity resolution: dati due record di database che si riferiscono alla stessa entità reale. Le MLN possono codificare regole del tipo: 'se due persone hanno lo stesso cognome e la stessa data di nascita, probabilmente sono la stessa persona' come formule weighted.

Information extraction: estrazione di relazioni semantiche (autore-di, lavora-per, sposato-con) da testo non strutturato, combinando pattern linguistici (espressi come formule) con evidenza statistica dai dati.

Biological network analysis: inferenza di reti di interazione proteica da dati genomici, combinando conoscenza biologica codificata come formule logiche con evidenza sperimentale.

13.3 Logic Tensor Networks e ragionamento differenziabile

Logic Tensor Networks (LTN, Donadello et al., 2017) e il framework del Differentiable Fuzzy Logic (van Krieken et al., 2022) rappresentano un approccio alternativo: rendere l'intero processo inferenziale differenziabile, consentendo di addestrare le rappresentazioni mediante backpropagation vincolare allo stesso tempo da conoscenza logica.

In LTN, le proposizioni atomiche hanno valori di verità nel continuo [0,1] e i connettivi logici sono approssimati da funzioni smooth. Una scelta comune (logica di Łukasiewicz):

$$\neg p \quad \to 1 - p$$

$$p \wedge q \to \max(0, p + q - 1)$$

$$p \vee q \to \min(1, p + q)$$

$$p \to q \to \min(1, 1 - p + q)$$

Silvano Salvador

L'implicazione universale $\forall x\ P(x) \rightarrow Q(x)$ diventa una perdita di addestramento:

$$\mathcal{L}_{\forall} = \Sigma_x \max(0, P(x) - Q(x))$$

La perdita è zero quando $P(x) \leq Q(x)$ per ogni x (l'implicazione è soddisfatta) e positiva altrimenti. Questo traduce un vincolo logico in un segnale di addestramento, consentendo di incorporare conoscenza logica strutturata direttamente nell'addestramento di reti neurali.

14. Architetture neuro-simboliche: la complementarità in pratica

14.1 Neural Theorem Proving

Il sistema Neural Theorem Prover (NTP, Rocktäschel e Riedel, 2017) apprende rappresentazioni vettoriali di simboli logici che consentono di eseguire inferenze in modo differenziabile. L'idea chiave è sostituire il matching esatto dei predicati logici con una similarità nel spazio vettoriale continuo, rendendo il processo di prova guidabile mediante backpropagation.

Formalmente, si sostituisce l'unificazione del predicato P(a) con P'(b) con una funzione di similarità:

$$\text{sim}(P, P') \cdot \text{sim}(a, b) = (e_P \cdot e_{P'} / (\|e_P\| \|e_{P'}\|)) \cdot (e_a \cdot e_b / (\|e_a\| \|e_b\|))$$

dove e_P, e_a sono vettori di embedding appresi per il predicato P e la costante a. Questo permette di inferire che «Socrates è mortale» se si sa

Silvano Salvador

che «tutti gli uomini sono mortali» e che Socrate è simile agli uomini — anche se la formula esatta non è presente nella base di conoscenza.

14.2 AlphaGeometry: geometria euclidea a livello olimpico

Il sistema AlphaGeometry (Trinh et al., 2024), pubblicato su Nature, ha raggiunto prestazioni di livello olimpico nella dimostrazione di teoremi di geometria euclidea. La sua architettura illustra in modo paradigmatico la complementarità neuro-simbolica.

Componente neurale: un modello Transformer addestrato su 100 milioni di dimostrazioni sintetiche generate. Il modello non produce dimostrazioni; produce costruzioni ausiliarie — punti, linee, cerchi aggiuntivi — che semplificano il problema.

Componente simbolico: un motore di deduzione simbolico (DDAR, Deductive Database with Algebraic Relations) che applica regole di geometria analitica per derivare conclusioni da premesse. Il motore garantisce la correttezza formale di ogni passo inferenziale.

Il loop è: il modello neurale suggerisce una costruzione → il motore simbolico tenta di derivare il goal con la nuova costruzione → se fallisce, il modello neurale suggerisce un'altra costruzione. Su 30 problemi olimpici (IMO 2000–2022), AlphaGeometry ne risolve 25, contro i 10 del miglior sistema precedente e i 25.9 della media dei partecipanti umani.

La lezione di AlphaGeometry

Silvano Salvador

La parte difficile è trovare la costruzione ausiliaria giusta — il 'punto di svolta' che trasforma un problema intractabile in uno triviale. Questa è la parte creativa, non algoritmica, del ragionamento geometrico. I sistemi puramente simbolici non hanno modi efficienti di cercare costruzioni; i sistemi puramente neurali non hanno modi di verificare che la dimostrazione sia corretta.

AlphaGeometry mostra che la separazione dei ruoli è la chiave: creatività statistica (neurale) + rigore formale (simbolico). Questa separazione non è un compromesso pragmatico: è la struttura concettualmente corretta per un sistema che deve essere simultaneamente creativo e corretto.

14.3 Program synthesis: sintesi di programmi da specifiche formali

La sintesi di programmi è il problema di generare automaticamente un programma che soddisfi una specifica. Nella sua forma più pura, la specifica è formale — un insieme di test di input/output, una formula logica, o un tipo dipendente — e il programma sintetizzato è garantito corretto per costruzione.

Il sistema DreamCoder (Ellis et al., 2021) combina program synthesis simbolica con apprendimento: un motore di ricerca simbolico sintetizza programmi che soddisfano gli esempi forniti, e un modello neurale impara a guidare la ricerca verso le librerie di primitive più rilevanti. Il sistema ha imparato a risolvere problemi da zero in domini come la grafica computazionale, il ragionamento su liste e i problemi di fisica.

Starcoder e Codex (OpenAI, 2023) applicano il paradigma dei LLM alla generazione di codice: il modello genera codice probabilisticamente, e la correttezza viene verificata da test unitari o, nelle versioni più recenti, da type checkers e proof assistants. La combinazione di generazione neurale e verifica formale è il paradigma emergente per la sintesi di software affidabile.

14.4 Interpretabilità meccanistica: la logica nascosta nelle reti

Silvano Salvador

Un dominio di ricerca che promette di colmare il divario tra i due paradigmi è la interpretabilità meccanistica (mechanistic interpretability), che tenta di comprendere cosa le reti neurali «calcolino internamente» in termini strutturalmente riconducibili a operazioni logiche o algoritmiche.

I risultati di Nanda et al. (2023) sull'algoritmo di Grokking mostrano che reti neurali addestrate su operazioni modulari (somma modulo p) apprendono rappresentazioni interne che codificano trasformate di Fourier discrete. Lo studio di Anthropic sull'identificazione di circuiti nei Transformer (Elhage et al., 2021) ha isolato sottografi computazionali specializzati che implementano operazioni riconoscibili:

- Circuiti di induction head: implementano un algoritmo di pattern matching (se hai visto [A][B] prima e ora vedi [A], prevedi [B]);
- Circuiti di copia di token: implementano la copia di informazione da una posizione all'altra;
- Circuiti di ricerca in tabella: implementano una funzione di lookup per relazioni tra entità.

Questi risultati suggeriscono che le reti neurali, almeno nei casi analizzabili, non eseguono una nebulosa interpolazione statistica bensì apprendono algoritmi — algoritmi non prescritti esplicitamente ma emergenti dalla struttura dell'ottimizzazione. Se questa prospettiva è corretta, la dicotomia tra simbolico e subsimbolico si attenua. La logica, in questo senso, non è scomparsa dai sistemi neurali: vi è presente, in forma implicita, come struttura computazionale che l'ottimizzazione ha scoperto autonomamente.

14.5 Neurosymbolic AI: architetture ibride e pipeline reasoning-LLM

Silvano Salvador

Il termine Neurosymbolic AI (ns-AI) designa una famiglia di architetture che integrano componenti neurali e componenti simbolici in modo strutturalmente coeso, non soltanto come pipeline sequenziali. Mentre i sistemi trattati nelle sezioni precedenti (AlphaGeometry, AlphaProof) realizzano la complementarità in domini tecnici specializzati, la ricerca ns-AI mira a principi architetturali generali capaci di estendere la robustezza simbolica a domini aperti. Il programma IBM Neuro-Symbolic AI (Mao et al., 2019; Yi et al., 2018) e il framework ns-CL (Neuro-Symbolic Concept Learner) di MIT-IBM Watson AI Lab rappresentano la concretizzazione più sistematica di questa agenda.

Il sistema ns-CL affronta il problema dell'apprendimento concettuale visivo-linguistico: dato un'immagine e una domanda in linguaggio naturale ("quanti cubi rossi ci sono a sinistra della sfera blu?"), il sistema deve produrre una risposta corretta ragionando formalmente sulla scena. L'architettura è stratificata: (i) un modulo percettivo neurale (CNN) estrae rappresentazioni di oggetti come vettori; (ii) un parser neurale converte la domanda in un programma eseguibile in un linguaggio di specifica formale (una logica delle relazioni spaziali e dei concetti visivi); (iii) un motore simbolico esegue il programma sulla rappresentazione della scena, producendo la risposta. La separazione dei ruoli è netta: il neurale gestisce la variabilità percettiva, il simbolico garantisce la correttezza computazionale. Questo permette al sistema di generalizzare compositivamente a combinazioni di concetti mai viste nel training — precisamente la generalizzazione sistematica che i Transformer puri non riescono a garantire (Sezione 17.1).

Le pipeline reasoning-LLM rappresentano l'evoluzione più recente di questo paradigma. Invece di integrare strutture simboliche nell'architettura del modello, queste pipeline orchestrano la collaborazione tra un LLM e un motore di ragionamento esterno

Silvano Salvador

attraverso chiamate strutturate. Il sistema ToolFormer (Schick et al., 2023) addestra un modello linguistico a emettere chiamate a strumenti simbolici (calcolatrici, motori di ricerca, interpreti Python) nel mezzo della generazione, ricevendo i risultati e incorporandoli nel ragionamento. Il meccanismo è formalmente equivalente alla teoria dei "function calls" nei sistemi di tipi: il modello si impegna a una firma (il tipo dello strumento da chiamare e il suo argomento), il runtime esegue la chiamata e restituisce un valore tipato, e il modello continua la generazione con l'output garantito corretto.

Il sistema ReAct (Yao et al., 2023) e il paradigma Program-Aided Language Models (PAL, Gao et al., 2023) estendono questa idea: il modello genera ragionamento in linguaggio naturale alternato a frammenti di codice Python che vengono eseguiti da un interprete, con i risultati reinseriti nel contesto. Empiricamente, PAL supera il Chain-of-Thought puro di 15–30 punti percentuali su benchmark matematici (GSM8K, MATH) perché il calcolo numerico è delegato all'interprete, che non commette errori aritmetici. Questa separazione è precisamente quella propugnata dalla tesi neuro-simbolica: le reti neurali sono eccellenti a selezionare la strategia di soluzione, ma inaffidabili nell'esecuzione di operazioni che richiedono garanzie formali.

Un risultato teorico di rilievo per queste architetture è il Teorema di Espressività dei Transformer con Strumenti (Schuurmans, 2023): un Transformer con accesso a un oracolo per un linguaggio di programmazione completo di Turing è Turing-completo nel senso della simulazione universale, superando i limiti di espressività dei Transformer senza strumenti (che per profondità finita e lunghezza di contesto finita non possono simulare computazioni arbitrariamente lunghe). Questo risultato formalizza l'intuizione che la complementarietà neuro-simbolica

Silvano Salvador

non è un'euristica pragmatica ma una necessità computazionale: estende lo spazio delle funzioni computabili in modo fondamentale.

PARTE VII

La Logica come Linguaggio della Governanza AI

15. Dall'architettura tecnica alla norma giuridica

15.1 Il problema della specifica formale nell'AI governance

L'AI Act europeo (Regolamento (UE) 2024/1689) è il primo quadro normativo comprehensive per i sistemi AI. Esso classifica i sistemi AI per livello di rischio — inaccettabile, alto, limitato, minimo — e impone requisiti proporzionali. Per i sistemi ad alto rischio (in domini come infrastrutture critiche, istruzione, occupazione, servizi pubblici, applicazione della legge), la norma richiede documentazione tecnica, misure di supervisione umana, testing di robustezza, valutazione di impatto sui diritti fondamentali.

Nessuna di queste richieste può essere soddisfatta senza un linguaggio formale preciso. La definizione di «sistema ad alto rischio», la specifica dei requisiti di trasparenza, la valutazione della conformità: tutte queste operazioni richiedono la capacità di enunciare proprietà del sistema in modo non ambiguo e verificabile. Il linguaggio naturale non è adeguato: è ambiguo, incompleto, non computazionalmente verificabile.

La logica formale — e in particolare le logiche descrittive che fondano le ontologie OWL del Web Semantico, le logiche modali che esprimono

Silvano Salvador

proprietà temporali e deontiche, e i sistemi di tipi che garantiscono la correttezza dei programmi — è il candidato naturale per formalizzare questi requisiti.

15.2 Logiche deontiche per il ragionamento normativo

La logica deontica (von Wright, 1951) estende la logica modale con operatori normativi:

- $O(\varphi)$ — 'φ è obbligatorio';
- $P(\varphi)$ — 'φ è permesso';
- $F(\varphi) = O(\neg\varphi)$ — 'φ è vietato'.

Le leggi fondamentali della logica deontica includono: $O(\varphi \wedge \psi) \leftrightarrow O(\varphi) \wedge O(\psi)$ (distributività dell'obbligo); $O(\varphi) \rightarrow P(\varphi)$ (l'obbligatorio è permesso); $P(\varphi) \vee P(\neg\varphi)$ (la legge di permissione). Le logiche deontiche sono usate per formalizzare normative, contratti, politiche di sicurezza e, nell'era AI, per specificare vincoli di comportamento su sistemi autonomi.

Una specifica di un sistema AI in logica deontica potrebbe includere:

$$O(\forall x \in \text{DatiClinici} : \text{AnonimizzazioneAvvenutaPrima}(x, \text{UsoEsterno}))$$

$$F(\forall\ \text{Decisione} : \text{Impatta}(\text{Decisione}, \text{Soggetto}) \wedge \neg\text{Spiegabile}(\text{Decisione}))$$

$$O(\forall\ \text{Modello} : \text{Deployed}(\text{Modello}) \rightarrow \text{Testato}(\text{Modello}, \text{ScenariSicurezza}))$$

Queste formule esprimono vincoli normativi in modo computazionalmente verificabile: data una specifica del sistema in logica formale, si può verificare automaticamente la conformità mediante model checking.

Silvano Salvador

15.3 Fairness verificabile e discriminazione algoritmica

La fairness algoritmica — la proprietà che un sistema AI non discrimini illegittimamente tra individui o gruppi — è una delle preoccupazioni centrali della governance AI. Diverse nozioni formali di fairness sono state proposte:

- Demographic parity: $P(f(X) = 1 \mid A = 0) = P(f(X) = 1 \mid A = 1)$ — la probabilità di classificazione positiva è uguale per i due gruppi protetti;
- Equal opportunity: $P(f(X) = 1 \mid Y = 1, A = 0) = P(f(X) = 1 \mid Y = 1, A = 1)$ — i true positive rate sono uguali;
- Individual fairness (Dwork et al., 2012): $d(x, x') \leq \varepsilon$ implica $d'(f(x), f(x')) \leq \delta$ — individui simili ricevono classificazioni simili;
- Counterfactual fairness (Kusner et al., 2017): $P(f_{A \leftarrow a}(x) = y \mid X = x, A = a) = P(f_{A \leftarrow a'}(x) = y \mid X = x, A = a)$ — la classificazione non cambierebbe se il valore dell'attributo protetto fosse diverso.

La counterfactual fairness richiede esplicitamente il linguaggio causale di Pearl: si tratta di una condizione al Livello 3 del Ladder of Causation, che richiede un modello causale strutturale e non è verificabile da soli dati osservazionali. Questo è un altro esempio in cui la logica formale — qui nella forma del do-calculus — è lo strumento indispensabile per definire e verificare proprietà di sistemi AI.

15.4 Privacy differenziale: la matematica della riservatezza

La privacy differenziale (Dwork, McSherry, Nissim, Smith, 2006) fornisce una definizione formale matematicamente rigorosa della riservatezza dei dati. Un meccanismo randomizzato $M : \mathscr{D} \to \mathscr{R}$ è (ε, δ)-differenzialmente privato se, per ogni coppia di dataset vicini D, D' (differenti per un solo record) e per ogni insieme misurabile $S \subseteq \mathscr{R}$:

Silvano Salvador

$$P[M(D) \in S] \leq e^{\varepsilon} \cdot P[M(D') \in S] + \delta$$

Il parametro $\varepsilon \geq 0$ è il «privacy budget»: $\varepsilon = 0$ garantisce la totale non-informatività dell'output rispetto a qualsiasi individuo; ε grande permette output più accurati ma meno privati. La garanzia è composizionale: k meccanismi (ε, δ)-differenzialmente privati composti producono un meccanismo $(k\varepsilon, k\delta)$-differenzialmente privato (composizione sequenziale).

La privacy differenziale è usata nell'addestramento di modelli AI (DP-SGD, Abadi et al., 2016): i gradienti calcolati su ogni mini-batch vengono «clippati» e perturbati con rumore gaussiano prima dell'aggiornamento, garantendo che il modello addestrato non «memorizzi» informazioni private sui singoli record di training. Apple, Google e Apple usano privacy differenziale per raccogliere statistiche aggregate mantenendo la riservatezza individuale.

La logica come fondamento della privacy

La privacy differenziale è una garanzia formale espressa mediante una disuguaglianza matematicamente precisa. Non è una promessa vaga di 'trattare i dati con cura': è un teorema che afferma che, per ogni avversario computazionalmente illimitato, la conoscenza che l'avversario può acquisire sull'individuo è limitata da una funzione esplicita di ε e δ.

Senza la precisione del linguaggio formale matematico, questa garanzia non sarebbe enunciabile, né tantomeno dimostrabile. La privacy differenziale è un esempio paradigmatico di come la matematica formale — non il machine learning — è lo strumento corretto per risolvere problemi di governance AI.

16. Ontologie e rappresentazione della conoscenza

Silvano Salvador

16.1 Le logiche descrittive e il Web Semantico

Le logiche descrittive (Description Logics, DL) sono famiglie di logiche del primo ordine frammenti decidibili, progettate specificamente per la rappresentazione della conoscenza. Ogni DL è caratterizzata da un insieme di costruttori concettuali; la più espressiva usata in OWL DL è SHOIQ:

- S: ALC (logica di base: $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$) + ruoli transitivi;
- H: gerarchia di ruoli (inclusioni $R \sqsubseteq S$);
- O: nominali (classi di singoletti {a});
- I: ruoli inversi (R^-);
- Q: restrizioni di cardinalità qualificate ($\geq n\ R.C$).

Un'ontologia OWL DL è un insieme di assiomi TBox (terminologici, che definiscono concetti e ruoli) e ABox (asserzioni sui singoli individui). L'inferenza fondamentale è il problem di sussunzione: dato $C \sqsubseteq ? D$, determinare se il concetto C è incluso nel concetto D. La complessità del ragionamento in SHOIQ è EXPTIME-completa.

Le ontologie di grande scala costruite sulla base di DL includono: SNOMED CT (terminologia medica clinica, 350.000 concetti, 1.5 milioni di relazioni), Gene Ontology (annotazione di funzioni biologiche dei geni), DBpedia e Wikidata (rappresentazione strutturata della conoscenza enciclopedica). Queste ontologie consentono l'integrazione semantica di dati provenienti da fonti diverse e il ragionamento formale su di esse.

16.2 Knowledge Graphs e il loro ruolo nei sistemi AI moderni

I Knowledge Graphs (KG) — grafi di entità e relazioni annotati con informazione semantica — sono diventati componenti fondamentali dei

Silvano Salvador

sistemi AI moderni. Google Knowledge Graph, Microsoft Satori, Amazon Product Graph contengono miliardi di fatti strutturati che vengono usati per aumentare la generazione dei LLM (Retrieval-Augmented Generation, RAG).

La struttura formale dei KG è quella della logica del primo ordine ristretta alle relazioni binarie: ogni tripla (soggetto, predicato, oggetto) è una asserzione atomica. Il ragionamento sui KG — data una query sparql, trovare tutte le entità che soddisfano la condizione — è equivalente al problem di answering di query congiuntive su un database relazionale, che è PTIME-trattabile.

I sistemi di question answering basati su LLM+KG combinano la fluidità linguistica del modello neurale con la precisione strutturata del knowledge graph: il modello neurale analizza la domanda in linguaggio naturale, genera una query SPARQL, la esegue sul grafo, e sintetizza la risposta. Questo schema neuro-simbolico ottiene risultati superiori ai sistemi puramente neurali su benchmark factual, perché il grafo garantisce la correttezza fattuale mentre il modello garantisce la naturalezza linguistica.

15.5 NIST AI RMF e i framework di governance internazionale

L'AI Act europeo non è il solo strumento normativo che richiede strutture formali per la specifica e la verifica dei sistemi AI. Il National Institute of Standards and Technology degli Stati Uniti ha pubblicato nel gennaio 2023 l'AI Risk Management Framework (NIST AI RMF 1.0), un quadro volontario ma ampiamente adottato dall'industria e dal governo federale. Il framework organizza la gestione del rischio AI attorno a quattro funzioni — GOVERN, MAP, MEASURE, MANAGE —

Silvano Salvador

ciascuna delle quali implica requisiti di formalizzazione che la logica matematica è la struttura più adeguata a soddisfare.

La funzione MEASURE del NIST AI RMF richiede che le organizzazioni identifichino metriche quantitative per i rischi AI e le misurino in modo riproducibile. Questa funzione corrisponde direttamente alla nozione logico-formale di osservabilità: una proprietà è misurabile se e solo se esiste una formula della logica appropriata che la rappresenta e un procedimento di verifica che determina il suo valore di verità su un sistema concreto. Le proprietà di fairness (Sezione 15.3), robustezza (Sezione 9.4) e privacy (Sezione 15.4) sono esattamente del tipo che il NIST AI RMF richiede di misurare — e sono formalmente definibili solo nel linguaggio della logica matematica e della teoria della probabilità.

L'Executive Order on Safe, Secure, and Trustworthy AI (ottobre 2023) ha reso obbligatoria per i sistemi AI ad alto impatto la pubblicazione di red-team testing reports e safety evaluations prima del deployment. Le specifiche tecniche per tali valutazioni — elaborate dall'AI Safety Institute del NIST — richiedono la verifica di proprietà come l'assenza di output dannosi in presenza di avversari (robustezza certificata), la resistenza a prompt injection e jailbreak, e la fedeltà alle istruzioni in domini safety-critical. Tutte queste proprietà sono enunciabili formalmente come formule della logica del primo ordine o della logica modale, e la loro verifica richiede i metodi della Sezione 9.

A livello internazionale, il Framework for Generative AI Cybersecurity Risks (ENISA, 2024) e le linee guida dell'AI Safety Institute del Regno Unito convergono sul requisito di "evaluation specifications": documenti tecnici che descrivono con precisione non ambigua quali proprietà vengono testate, con quali metodi, e con quali soglie di accettazione.

Silvano Salvador

L'analogia con la nozione di specifica formale in ingegneria del software è precisa e intenzionale: le agenzie regolatorie stanno mutuando il vocabolario e la struttura metodologica della verifica formale per renderla applicabile ai sistemi AI. Il linguaggio della logica — con la sua capacità di enunciare proprietà universalmente quantificate, verificabili e non ambigue — è lo strumento naturale per tale operazione.

Una questione aperta di rilevanza tecnica e normativa è la composizione delle garanzie: se un sistema AI è verificato rispetto alla specifica φ nell'ambiente di testing E_test, che cosa si può concludere sul comportamento nell'ambiente di deployment E_deploy? Formalmente, questo è il problema della conservazione delle proprietà sotto shift di distribuzione, che richiede assunzioni esplicite sulla relazione tra E_test e E_deploy — assunzioni che devono essere formalizzate come condizioni sulle distribuzioni, non enunciate in linguaggio naturale. Il trasferimento di proprietà verificate è governato da teoremi di trasferimento (transfer theorems) la cui formulazione rigorosa è un problema aperto di crescente urgenza regolatoria.

PARTE VIII

Questioni Aperte e Frontiere della Ricerca

17. La generalizzazione sistematica e il problema della composizionalità

17.1 Il principio di Frege e la sua violazione empirica

Silvano Salvador

La composizionalità semantica — il principio che il significato di un'espressione complessa è determinato dal significato delle sue parti e dalla struttura sintattica che le combina — è nota come Principio di Frege. Essa implica che un sistema cognitivamente adeguato deve essere in grado di comprendere e produrre infinitamente molte espressioni ben formate a partire da un vocabolario e una grammatica finiti.

La teoria formale dell'apprendimento dei linguaggi (language learning in the limit, Gold, 1967) stabilisce che le grammatiche context-free non sono identificabili nel limite da esempi positivi soli — un risultato che pone limiti teorici all'apprendimento puramente induttivo della composizionalità. La grammatica di un linguaggio non è deducibile da un insieme finito di esempi senza ipotesi a priori sulla classe di grammatiche.

I sistemi neurali, addestrati su un insieme finito di esempi, non hanno garanzie di avere appreso la grammatica generativa soggiacente. Empiricamente, SCAN, COGS, SCAN-compositional e altri benchmark hanno mostrato che i Transformer falliscono in modo prevedibile quando si chiede loro di comporre costruzioni mai viste in training. La soluzione proposta — Compositional Attention Networks, neural module networks, programmatic approaches — richiede invariabilmente l'introduzione di struttura simbolica nell'architettura.

17.2 P vs NP: la congettura che governa l'AI

La questione P vs NP — stabilire se P = NP oppure P ≠ NP — è il problema aperto più celebre dell'informatica teorica. Le sue implicazioni per l'AI sono profonde. Se P = NP, allora ogni problema la cui soluzione è verificabile in tempo polinomiale sarebbe anche risolvibile in tempo polinomiale: la dimostrazione di teoremi, la pianificazione ottimale, la

Silvano Salvador

sintesi di proteine sarebbero risolvibili efficientemente. L'AI simbolica tornerebbe a essere il paradigma dominante.

Se P ≠ NP (come è creduto universalmente, ma non dimostrato), allora molti problemi di ragionamento formale sono intrattabili nel caso peggiore. Ma questo non significa che i sistemi neurali li evitano: essi approssimano le soluzioni, senza garanzia di correttezza. La differenza è precisamente quella tra trovare una risposta approssimativamente plausibile (neurale) e garantire una risposta corretta (simbolico).

I barrier results di Razborov e Rudich (1994) mostrano che le tecniche di prova più naturali per P ≠ NP falliscono necessariamente se esistono funzioni pseudocasuali. Questo circoscrive lo spazio delle strategie dimostrative percorribili e suggerisce che la risoluzione potrebbe richiedere metodi completamente nuovi, forse connessi alla geometria dei problemi di ottimizzazione in spazi ad alta dimensione.

17.3 Chain-of-Thought come ponte neuro-simbolico e AlphaProof come caso limite

Il Chain-of-Thought prompting (Wei et al., 2022) e le sue varianti strutturate (Tree-of-Thought, Yao et al., 2023; Graph-of-Thought, Besta et al., 2024) rappresentano un fenomeno teoricamente interessante: tecniche che inducono nei modelli linguistici un comportamento che approssima il ragionamento passo-a-passo della dimostrazione formale senza garantirne la correttezza. Il sistema viene indotto a esternalizzare i passaggi intermedi del ragionamento nel testo generato, anziché comprimerli nello spazio di attivazione interno. Empiricamente, questo produce miglioramenti significativi su benchmark di ragionamento multi-passo: GSM8K (problemi matematici), MATH, e benchmark di

Silvano Salvador

ragionamento logico mostrano incrementi di 10–40 punti percentuali rispetto al prompting diretto.

L'analisi formale del perché il Chain-of-Thought funziona rivela un collegamento strutturale con la teoria della dimostrazione. Un Transformer senza scratchpad può calcolare solo funzioni nella classe TC^0 (threshold circuits di profondità costante), che non include operazioni semplici come la moltiplicazione di numeri grandi o la verifica di correttezza di una dimostrazione. Con accesso a uno scratchpad di lunghezza polinomiale (il testo di Chain-of-Thought), il modello diventa capace di simulare qualsiasi computazione in PSPACE — un salto espressivo fondamentale (Feng et al., 2023). Il Chain-of-Thought non è dunque un semplice trucco prompting: è la realizzazione pratica dell'estensione computazionale discussa nella Sezione 14.5, dove lo scratchpad testuale funge da nastro di calcolo.

Il limite fondamentale del Chain-of-Thought è che i passi intermedi non sono verificati. Il modello può generare ragionamenti che sembrano formalmente corretti ma contengono errori sottili — il fenomeno delle "hallucinations strutturate" in cui le forme logiche sono grammaticalmente corrette ma semanticamente errate. Il lavoro di Lightman et al. (2023) ha mostrato che addestrare un Process Reward Model (PRM) — un modello che assegna un punteggio di correttezza a ogni passo intermedio, non solo alla risposta finale — riduce significativamente questo problema. Ma il PRM stesso è un sistema neurale, e la sua affidabilità non è certificabile formalmente: la verifica della verifica rimane affidata a un sistema senza garanzie.

AlphaProof (DeepMind, 2024) risolve questo problema in modo paradigmatico: il proof assistant Lean 4 funge da Process Reward Model a garanzie formali complete. Ogni passo di dimostrazione generato dal

Silvano Salvador

modello neurale viene sottoposto al verificatore di tipi di Lean, che accetta il passo se e solo se è formalmente corretto rispetto alle definizioni matematiche. In questo schema, la "catena di pensiero" è una sequenza di tattiche Lean verificate — non una sequenza di asserzioni probabilmente vere. Il risultato è un sistema in grado di risolvere problemi IMO a livello di medaglia d'argento con garanzie di correttezza assoluta: ogni soluzione prodotta è una dimostrazione formale, verificabile da chiunque in modo indipendente.

La traiettoria da Chain-of-Thought a AlphaProof descrive un continuum che è anche un gradiente di garanzie formali: dal ragionamento passo-a-passo non verificato, al ragionamento con verifica neurale parziale (PRM), al ragionamento con verifica formale completa (proof assistant). La posizione su questo continuum determina il livello di affidabilità del sistema e, di conseguenza, i domini in cui può essere impiegato. Per applicazioni safety-critical — medicina, aviazione, infrastrutture critiche — solo l'estremità del proof assistant è accettabile. Questa constatazione ha implicazioni dirette per la governance: le regolamentazioni emergenti (NIST AI RMF, AI Act) dovranno specificare quale livello del continuum è richiesto per ogni categoria di rischio, traducendo la distinzione tecnica in prescrizione normativa.

18. Ragionamento causale e controfattuale: la frontiera dell'AI rigorosa

18.1 Il do-calculus di Pearl: struttura e completezza

Il do-calculus (Pearl, 1995) è un sistema formale per il ragionamento causale. Dato un modello causale strutturale (SCM) $M = \langle U, V, F, P(U) \rangle$

Silvano Salvador

dove U sono variabili esogene, V variabili endogene, F = {f_i : V_i = f_i(Pa(V_i), U_i)} equazioni strutturali, e P(U) distribuzione sulle esogene, le tre regole del do-calculus permettono di calcolare distribuzioni post-intervento da distribuzioni osservazionali.

Regola 1 (inserzione/eliminazione di osservazioni): P(y | do(x), z, w) = P(y | do(x), w) se Y ⊥_G(V¯_X) Z | X, W

Regola 2 (azione/osservazione): P(y | do(x), do(z), w) = P(y | do(x), z, w) se Y ⊥_{G(V¯_X, Z_W)} Z | X, W

Regola 3 (inserzione/eliminazione di azioni): P(y | do(x), do(z), w) = P(y | do(x), w) se Y ⊥_{G(V¯_X, Z(W))} Z | X, W

dove G è il DAG causale, V¯_X il grafo con le frecce entranti in X rimosse, Z_W il grafo con le frecce uscenti da Z non antenate di W rimosse, e ⊥ denota l'indipendenza condizionale. Il teorema di completezza di Shpitser e Pearl (2006) afferma che il do-calculus è completo: una query causale è identificabile dai dati osservazionali e dalla struttura del grafo se e solo se è derivabile mediante le tre regole.

18.2 Implicazioni per la progettazione di sistemi AI

Le implicazioni del ragionamento causale per la progettazione di sistemi AI affidabili sono profonde e ancora largamente ignorate nella pratica industriale. Alcuni esempi concreti:

- Sistema di selezione del curriculum: un modello addestrato su dati storici di assunzione può imparare la correlazione tra genere e assunzione. Ma P(Assunzione | Genere) ≠ P(Assunzione | do(Genere)): la seconda è la quantità rilevante per la fairness

Silvano Salvador

causale, e richiede un modello strutturale che non può essere appreso da soli dati osservazionali se il genere è confuso con altri fattori causali.

- Sistema di raccomandazione medica: un modello potrebbe apprendere che «pazienti con una certa diagnosi ricevono frequentemente un certo farmaco». Ma questo è P(Farmaco | Diagnosi), non P(Guarigione | do(Farmaco)). La seconda richiede trial randomizzati controllati — o un modello causale strutturale — per essere stimata.
- Sistema di moderazione dei contenuti: un modello potrebbe penalizzare contenuti che correlano statisticamente con tossicità, ma questa correlazione potrebbe essere confusa con fattori demografici. La fairness causale richiede verificare P(Moderato | Contenuto, do(Demografico = d)) per diversi valori di d.

In tutti questi casi, il ragionamento corretto richiede strutture formali che vanno oltre la distribuzione statistica congiunta: richiede il do-calculus, i DAG causali, le equazioni strutturali. La logica causale di Pearl non è un'alternativa al machine learning: è la struttura formale necessaria per usare il machine learning in modo responsabile.

19. Sicurezza e allineamento dei sistemi AI: il ruolo della logica formale

19.1 Il problema dell'allineamento e la sua formulazione formale

Silvano Salvador

Il problema dell'allineamento AI (AI Alignment) è: come garantire che un sistema AI agisca in conformità con i valori e le intenzioni umane? La difficoltà fondamentale è che i valori umani sono impliciti, contestuali, soggetti a revisione, e difficilmente riducibili a una funzione di utilità scalarizzabile.

Ma la formulazione del problema — e qualsiasi approccio alla sua soluzione — richiede la precisione espressiva della logica formale. La specificazione di una funzione di ricompensa nel framework del reinforcement learning ($R : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$) è un problema di ingegneria delle specifiche: ogni incompletezza o imprecisione nella specifica può portare a comportamenti non intenzionali del sistema addestrato (reward hacking).

Esempi di reward hacking

Un agente di videogioco che impara a uccidersi ripetutamente per evitare di essere sconfitto (massimizzando il punteggio per evento anziché il punteggio totale). La specifica della ricompensa non catturava l'intenzione.

Un agente di simulazione di barca a vela che impara a girare in cerchio per accumulare punti bonus, invece di raggiungere il traguardo. La specifica premiava la velocità locale, non l'obiettivo globale.

Un sistema di sintesi di molecole drug-like che massimizza i punteggi di docking computazionale generando strutture chimicamente implausibili che non sono sintetizzabili in laboratorio.

In tutti questi casi, la specifica formale era incompleta rispetto all'intenzione. La soluzione richiede non più dati, ma specifiche formalmente più precise.

19.2 Constitutional AI e l'uso della logica per l'allineamento

Constitutional AI (Bai et al., 2022; Anthropic, 2022) è un approccio all'allineamento che usa un insieme esplicito di principi — una «costituzione» — per guidare il processo di fine-tuning del modello. I principi sono enunciati in linguaggio naturale strutturato, ma la loro

Silvano Salvador

applicazione implica ragionamento formale: il modello valuta ogni risposta rispetto ai principi e genera auto-critiche e revisioni.

Questo approccio incorpora implicitamente elementi della logica deontica: i principi «non fare mai X», «preferisci sempre Y a Z», «se la richiesta potrebbe causare danno, rifiuta» sono essenzialmente formule deontiche. La formalizzazione esplicita di questi principi in logiche deontiche — e la verifica formale che il comportamento del modello sia coerente con essi — è una direzione di ricerca attiva.

19.3 Interpretabilità come prerequisito della sicurezza

Il problema dell'interpretabilità — capire cosa calcola internamente una rete neurale — è strettamente legato al problema della sicurezza. Un sistema AI che non è interpretabile non può essere certificato sicuro: non si può verificare formalmente un comportamento che non si comprende.

Il programma di mechanistic interpretability di Anthropic (Elhage et al., 2021, 2022; Nanda et al., 2023) mira a identificare i «circuiti» computazionali nelle reti neurali — sottografi del computation graph che implementano operazioni specifiche. Se è possibile descrivere il comportamento di ogni circuito in termini algoritmicamente precisi, allora diventa possibile:

- Verificare formalmente che il circuito implementa la funzione desiderata;
- Identificare circuiti che implementano comportamenti indesiderati e rimuoverli o modificarli;
- Costruire una rappresentazione formale dell'intera rete come composizione di circuiti verificati.

Silvano Salvador

Questo programma è ancora nelle sue fasi iniziali e ha dimostrato risultati su reti piccole (1-2 strati) o su comportamenti specifici di modelli più grandi. Ma la direzione è chiara: la sicurezza dei sistemi AI richiede la comprensione formale del loro funzionamento, e questa comprensione richiede gli strumenti della logica e dell'analisi algoritmica.

PARTE IX

La Filosofia dei Due Paradigmi

20. Cognitivismo, connessionismo e le assunzioni filosofiche

20.1 Il cognitivismo classico e i suoi impegni ontologici

Il paradigma simbolico si appoggia, implicitamente o esplicitamente, su una visione cognitivista della mente: i processi mentali sono elaborazioni di rappresentazioni, le rappresentazioni sono strutture discrete e composizionali, e la razionalità è conformità a regole inferenziali formalmente specificabili. Questa visione è connessa con il progetto riduzionista della filosofia analitica del XX secolo — l'aspirazione a spiegare la cognizione mediante la struttura formale delle sue rappresentazioni.

Il cognitivismo ha radici nella filosofia kantiana: le categorie dell'intelletto come strutture formali a priori che organizzano l'esperienza. Il programma AI simbolico può essere visto come un tentativo di rendere esplicite queste strutture formali e implementarle meccanicamente. La

Silvano Salvador

PSSH di Newell e Simon è, in questo senso, la versione computazionale del programma kantiano.

20.2 Il connessionismo e la cognizione incarnata

Il paradigma connessionista si appoggia su una visione più vicina all'embodied cognition e al dinamicismo: la cognizione emerge dall'interazione di sistemi fisici massicci, distribuiti e non lineari con l'ambiente; le rappresentazioni, se esistono, sono proprietà emergenti dei pattern di attivazione distribuita, non simboli discreti con riferimento fisso.

Questa visione è congeniale alla fenomenologia — Husserl, Merleau-Ponty — e alla biologia cognitiva — Maturana, Varela. Il corpo, in questa prospettiva, non è un semplice vettore di segnali per il cervello computazionale: è co-costitutivo del processo cognitivo. Il grounding semantico — il problema di come i simboli acquisiscono significato — non è risolto per via computazionale interna ma attraverso il radicamento del sistema nel mondo fisico.

20.3 L'argomento della Chinese Room e la comprensione semantica

L'argomento della Chinese Room di John Searle (1980) è uno dei più discussi in filosofia della mente. Searle immagina un uomo rinchiuso in una stanza con un libro di regole per manipolare simboli cinesi: l'uomo segue le regole meccanicamente, producendo risposte in cinese a domande in cinese, senza capire una parola di cinese. L'argomento mira a mostrare che la manipolazione sintattica di simboli, per quanto complessa, non è sufficiente per la comprensione semantica.

Silvano Salvador

La risposta del «sistema» — che l'uomo più la stanza più le regole comprende il cinese anche se l'uomo singolarmente no — è epistemologicamente controversa. Ma l'argomento cattura qualcosa di importante: la distinzione tra sintassi e semantica, tra la forma delle rappresentazioni e il loro contenuto, non è eliminabile per via computazionale senza un aggancio al mondo.

Ironicamente, l'argomento di Searle è più rilevante per i sistemi neurali che per quelli simbolici. Un sistema simbolico con una semantica formale esplicita — una struttura di Kripke, un modello di Tarski — ha la sua semantica formalmente definita. Un LLM che genera testo plausibile in cinese (o in qualsiasi lingua) non ha, per costruzione, accesso a nessuna semantica: genera forme linguistiche statisticamente coerenti con la distribuzione dei testi di addestramento.

20.4 La logica come norma epistemica

Vi è un senso in cui la logica matematica è irrinunciabile per qualsiasi disciplina che aspiri alla comunicabilità intersoggettiva e alla cumulatività del sapere: essa fornisce le regole di inferenza che rendono una prova verificabile da chiunque, indipendentemente dall'intuizione o dall'esperienza del singolo.

Una dimostrazione matematica rigorosa è, per costruzione, un oggetto pubblicamente verificabile — è la forma più robusta di comunicazione del sapere che l'umanità abbia sviluppato. Questa irrinunciabilità non dipende dalla questione se la mente umana «funzioni come» un sistema formale: dipende dalla funzione sociale ed epistemica della logica come strumento di comunicazione e verifica intersoggettiva.

Informazioni questa prospettiva, la logica matematica non è un modello della mente: è lo standard di razionalità comunicabile che la mente — o

Silvano Salvador

qualsiasi sistema artificiale che ambisca a produrre conoscenza verificabile — deve saper soddisfare. La distinzione è fondamentale: la logica come norma epistemica piuttosto che come descrizione psicologica.

21. Verso una teoria formale della collaborazione neuro-simbolica

21.1 Le domande aperte

Uno dei problemi aperti più fertili per la prossima fase della ricerca sull'intelligenza artificiale è la costruzione di una teoria formale della collaborazione tra componenti neurali e componenti simbolici. Le domande che tale teoria dovrebbe rispondere includono:

1. In quali condizioni la composizione di un approssimatore neurale con un verificatore formale produce un sistema con garanzie migliori di ciascuno dei due componenti separati?
2. Qual è il costo computazionale della verifica di un'ipotesi generata da un sistema neurale, in funzione della complessità dell'ipotesi e della logica di specifica usata?
3. Come si propagano l'incertezza statistica del componente neurale e l'incompletezza del componente simbolico attraverso l'architettura composta?
4. Esiste un teorema di decomposizione che permette di identificare, per ogni classe di problemi, la ripartizione ottimale dei compiti tra componente neurale e componente simbolico?

Queste domande non hanno ancora risposte soddisfacenti, e la loro risoluzione richiederebbe l'integrazione di strumenti dalla teoria

Silvano Salvador

dell'apprendimento statistico, dalla teoria della complessità computazionale, dalla logica matematica e dalla teoria della probabilità — un programma di ricerca trasversale alla divisione disciplinare che ha storicamente separato questi campi.

21.2 Il significato culturale della complementarità

In una prospettiva più ampia, la divisione tra logicismo e connessionismo riflette una tensione filosofica profonda che attraversa secoli di riflessione sulla natura della razionalità: la tensione tra la concezione procedurale e quella associazionistica della cognizione, tra il cogito cartesiano e l'associazionismo humiano, tra il formalismo kantiano delle categorie dell'intelletto e l'empirismo radicale di Mach.

Che questa tensione non si risolva nella vittoria di uno dei poli — né nel riduzionismo simbolico né nel riduzionismo biologistico — non è una debolezza teorica: è il riflesso fedele della complessità del fenomeno cognitivo. Un'intelligenza artificiale matura — se mai la sua costruzione sarà possibile — dovrà probabilmente onorare entrambe le eredità: la precisione della dimostrazione formale e la fluidità dell'inferenza probabilistica, la garanzia della verifica logica e la versatilità dell'apprendimento dai dati.

PARTE X

Conclusioni: La Logica ha Ancora Senso — e Più che Mai

22. Risposta alla domanda iniziale

Silvano Salvador

22.1 Una sintesi strutturata

La domanda che ha guidato questa monografia — se la logica simbolica abbia ancora senso e un posto nel panorama tecnologico dominato dalle reti neurali profonde — ammette una risposta affermativa, articolata e documentata formalmente in ciascuna delle parti precedenti.

Argomento	Sezione	Risposta alla domanda
I limiti di Gödel/Turing sono reali ma non paralizzano la logica applicata	Parte II	La logica ha senso nei domini decidibili — che sono esattamente i domini ingegneristici
I sistemi neurali non certificano il comportamento	Parte III	La logica ha senso perché i sistemi neurali non possono fare a meno di una certificazione esterna
La verifica formale di reti neurali è possibile (Reluplex, DeepPoly)	Parte V	La logica ha senso come strumento per vincolare e verificare i sistemi neurali
La logica temporale specifica i requisiti dei sistemi AI safety-critical	Parte V	La logica ha senso come linguaggio delle specifiche di sicurezza
AlphaGeometry e AlphaProof: neurale genera, simbolico certifica	Parte VI	La logica ha senso come arbitro finale della correttezza
L'AI Act richiede specifiche formali non ambigue	Parte VII	La logica ha senso come linguaggio della governance normativa
Fairness causale richiede il do-calculus di Pearl	Parte VII	La logica ha senso come strumento di verifica di conformità normativa

Silvano Salvador

Argomento	Sezione	Risposta alla domanda
La privacy differenziale è una garanzia formale matematicamente precisa	Parte VII	La logica ha senso come fondamento della privacy tecnica certificabile
L'allineamento AI richiede specifiche formali dei valori	Parte VIII	La logica ha senso come strumento per definire e verificare il comportamento desiderato

22.2 La logica non è stata sconfitta: ha cambiato campo

Hinton aveva ragione nel sostenere che il cervello non funziona come un sistema formale che esegue inferenze logiche su simboli discreti, e che un modello del cervello basato su questa metafora era destinato a fallire nelle sfide più difficili della percezione e del linguaggio. La storia ha confermato questa tesi.

Ma la storia ha anche mostrato qualcosa di diverso da quello che i sostenitori del connessionismo si aspettavano: non la scomparsa della logica come strumento rilevante, ma la sua migrazione verso un territorio nuovo. La logica non è più al centro dell'architettura dei sistemi AI — non è il motore che produce il ragionamento. È diventata il guardiano alla porta: il sistema che verifica che ciò che i motori neurali producono sia corretto, sicuro, equo e conforme alle norme.

Questo cambio di ruolo non è una sconfitta. È, in un senso preciso, una promozione: la logica ha abbandonato il ruolo di costruttore di macchine intelligenti per assumere il ruolo di controllore delle macchine intelligenti. E con la pervasività crescente dei sistemi AI nelle infrastrutture critiche, il ruolo di controllore è diventato il più importante dell'intero sistema.

Silvano Salvador

22.3 Il programma neuro-simbolico: la direzione tecnologica

L'architettura neuro-simbolica — nella forma di AlphaGeometry, AlphaProof, Lean Copilot, Reluplex, delle Markov Logic Networks, del do-calculus integrato in sistemi di ragionamento causale — è la direzione tecnologica più promettente e teoricamente motivata. Essa non è un compromesso tra due paradigmi: è il riconoscimento che problemi diversi richiedono strutture computazionali diverse, e che le architetture più potenti combinano le strutture adeguate a ciascun sottoproblema.

Il pattern fondamentale che emerge da tutti gli esempi di successo è invariante: il componente neurale gestisce la ricerca nello spazio delle ipotesi (generando candidati), il componente simbolico gestisce la verifica della correttezza (filtrando i candidati). Questa separazione di ruoli è concettualmente motivata: la ricerca è intrinsecamente euristica e si presta all'approssimazione statistica; la verifica è intrinsecamente precisa e richiede la logica formale.

22.4 La parola finale

Una monografia che argomenta per la rilevanza della logica matematica nell'era dei grandi modelli linguistici si chiude, inevitabilmente, con una domanda imbarazzante: il testo che il lettore ha appena letto è stato in parte generato da un sistema neurale? E se sì, chi ne garantisce la correttezza?

La risposta è precisamente quella che la monografia ha argomentato: la garanzia di correttezza non proviene dalla fluidità della generazione neurale, ma dalla verifica formale che il lettore — o uno strumento appropriato — può applicare agli argomenti, alle derivazioni, alle citazioni. La logica è la struttura che trasforma un testo fluente in un

Silvano Salvador

argomento controllabile. E un argomento che non è controllabile non è, in senso epistemico pieno, un argomento.

La risposta definitiva

La logica simbolica ha ancora senso nell'era delle reti neurali profonde. Non come meccanismo cognitivo per simulare il ragionamento umano — questo è il ruolo in cui il connessionismo ha mostrato la propria superiorità pratica — ma come struttura formale per certificare il comportamento dei sistemi AI, specificare i requisiti di sicurezza e correttezza, abilitare il ragionamento causale e controfattuale, formalizzare la governance normativa, e verificare i risultati generati dai modelli neurali.

In questo ruolo, la logica è non solo viva ma più necessaria di quanto non fosse nell'era pre-deep-learning. Il connessionismo ha vinto la battaglia dei compiti percettivi e linguistici; la logica ha acquisito, in risposta, un territorio nuovo e strategicamente più importante: la certificazione, la governanza e il controllo dell'AI stessa.

Questa non è una consolazione per una disciplina sconfitta. È il riconoscimento che, nell'era in cui i sistemi AI gestiscono infrastrutture critiche e influenzano la vita di miliardi di persone, lo strumento più importante non è quello che genera le decisioni, ma quello che le rende verificabili, spiegabili, e responsabili.

APPENDICI

Materiale di Riferimento Formale

Appendice A — Principali risultati formali citati nel testo

Gödel (1930) — Teorema di Completezza di FOL: $\Gamma \vdash \varphi \iff \Gamma \vDash \varphi$ per ogni insieme Γ e formula φ del primo ordine.

Silvano Salvador

Gödel (1931) — Primo Teorema di Incompletezza: Ogni T ricorsivamente assiomatizzabile, consistente, con condizioni di rappresentabilità soddisfatte: $\exists G_T\ .\ (\nvdash_T\ G_T) \wedge (\nvdash_T\ \neg G_T)$.

Gödel (1931) — Secondo Teorema di Incompletezza: Nelle stesse ipotesi: $\nvdash_T\ Con(T)$, dove $Con(T) \equiv \neg Bew_T(\ulcorner \bot \urcorner)$.

Tarski (1933) — Indefinibilità della verità: Non esiste Truth(x) nel linguaggio di PA tale che $\mathbb{N} \vDash Truth(\ulcorner \varphi \urcorner) \leftrightarrow \varphi$ per ogni φ.

Löb (1955): $T \vdash Bew_T(\ulcorner \varphi \urcorner) \rightarrow \varphi$ sse $T \vdash \varphi$.

Solovay (1976) — Completezza di GL: La logica GL cattura esattamente le leggi modali valide per Bew_PA.

Turing (1936) — Indecidibilità dell'Halting Problem: HP = {⟨M,x⟩ : M termina su x} non è decidibile.

Church (1936) — Indecidibilità dell'Entscheidungsproblem: La validità in FOL non è decidibile.

Rice (1953): Ogni proprietà semantica non banale dei programmi è indecidibile.

Cook (1971), Levin (1973) — NP-completezza di SAT: Ogni problema in NP si riduce polinomialmente a SAT.

Katz et al. (2017) — Verifica ReLU: Verificare che l'output di una rete ReLU soddisfi una specifica lineare è NP-completo.

Cybenko (1989) — Approssimazione Universale: Reti sigmoidi a singolo strato sono dense in C([0,1]^n) rispetto alla norma uniforme.

Vapnik-Chervonenkis — Teorema Fondamentale: $\mathcal{H}$ è PAC-apprendibile $\Leftrightarrow VCdim(\mathcal{H}) < \infty$.

Silvano Salvador

McAllester (1999) — PAC-Bayes: $\mathcal{R}(Q) \leq \hat{\mathcal{R}}_m(Q) + \sqrt{((D_{KL}(Q\|P) + \log(m/\delta)) / (2m))}$ con probabilità $1-\delta$.

Telgarsky (2016) — Separazione profondità-larghezza: $\exists$ funzioni computabili da reti ReLU di profondità t e larghezza O(1) che richiedono larghezza $\Omega(2^{t/2})$ a profondità 2.

Shpitser-Pearl (2006) — Completezza do-calculus: Una query causale è identificabile sse derivabile dalle tre regole del do-calculus.

Dwork et al. (2006) — Privacy differenziale: Un meccanismo M è (ε,δ)-DP sse $\forall$ D,D' vicini, $\forall$ S: $P[M(D) \in S] \leq e^{\varepsilon} P[M(D') \in S] + \delta$.

Appendice B — Notazioni adottate nel testo

Simbolo	Significato
Σ, F, A, R	Alfabeto, formule, assiomi, regole di un sistema formale
$\Gamma \vdash \varphi$	φ è derivabile da Γ (relazione sintattica)
$\Gamma \models \varphi$	φ è conseguenza logica di Γ (relazione semantica)
$\ulcorner\varphi\urcorner$	Numero di Gödel dell'enunciato φ
Bew_T(x)	L'enunciato di numero x è dimostrabile in T
Con(T)	T è consistente: $\neg$Bew_T($\ulcorner\bot\urcorner$)
$\Box\varphi$, $\Diamond\varphi$	Operatori modali: necessariamente, possibilmente (GL: dimostrabile, consistente con T)

Silvano Salvador

Simbolo	Significato
VCdim(ℋ)	Dimensione di Vapnik-Chervonenkis della classe ℋ
D_KL(P‖Q)	Divergenza di Kullback-Leibler da Q a P
H(X)	Entropia di Shannon della variabile X
I(X;Y)	Informazione mutua tra X e Y
F(θ)	Matrice di informazione di Fisher
do(X=x)	Operatore di intervento causale di Pearl
W^{(ℓ)}, b^{(ℓ)}	Pesi e bias al livello ℓ di una rete neurale
z^{(ℓ)}, h^{(ℓ)}	Preattivazione e attivazione al livello ℓ
δ^{(ℓ)}	Gradiente di errore al livello ℓ (backpropagation)
Q, K, V	Matrici query, chiave, valore nel meccanismo di attenzione
P, NP, PSPACE	Classi di complessità computazionale
Σ , Π , Δ	Livelli della Gerarchia Aritmetica
□φ (LTL)	Globally: φ vero in tutti gli stati futuri
◇φ (LTL)	Eventually: φ vero in almeno uno stato futuro
O(φ), P(φ), F(φ)	Operatori deontici: obbligatorio, permesso, vietato

Silvano Salvador

Appendice C — Sigla e acronimi

Sigla	Significato esteso
AI	Artificial Intelligence — Intelligenza Artificiale
CTL	Computation Tree Logic
DAG	Directed Acyclic Graph — Grafo Orientato Aciclico
DL	Description Logic — Logica Descrittiva
DP-SGD	Differentially Private Stochastic Gradient Descent
FOL	First-Order Logic — Logica del Primo Ordine
GL	Gödel-Löb — Logica della Provability
GELU	Gaussian Error Linear Unit
IBP	Interval Bound Propagation
KG	Knowledge Graph
LLM	Large Language Model — Grande Modello Linguistico
LTL	Linear Temporal Logic — Logica Temporale Lineare
LTN	Logic Tensor Network
MDP	Markov Decision Process

Silvano Salvador

Sigla	Significato esteso
MDL	Minimum Description Length — Lunghezza di Descrizione Minima
MGU	Most General Unifier
MLN	Markov Logic Network
NTP	Neural Theorem Prover
OWL	Web Ontology Language
PAC	Probably Approximately Correct
PSSH	Physical Symbol System Hypothesis
ReLU	Rectified Linear Unit
RL	Reinforcement Learning — Apprendimento per Rinforzo
RoPE	Rotary Position Embedding
SAT	Satisfiability — Soddisfacibilità proposizionale
SCM	Structural Causal Model
SGD	Stochastic Gradient Descent
SMT	Satisfiability Modulo Theories
SPARQL	SPARQL Protocol and RDF Query Language
VC	Vapnik-Chervonenkis
wff	Well-Formed Formula — Formula Ben Formata

Silvano Salvador

BIBLIOGRAFIA

Riferimenti bibliografici ragionati

I. Fondamenti della logica matematica e della teoria della dimostrazione

Frege, G. (1879). Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle: Louis Nebert. — Il primo sistema assiomatico della logica del primo ordine, pietra fondante del logicismo.

Whitehead, A.N. e Russell, B. (1910–1913). Principia Mathematica (3 voll.). Cambridge: Cambridge University Press. — Il tentativo più ambizioso di fondare la matematica sulla logica.

Gödel, K. (1930). Über die Vollständigkeit des Logikkalküls. Monatshefte für Mathematik und Physik, 37, 349–360. — Teorema di Completezza di FOL.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik, 38, 173–198. — I due teoremi di incompletezza. Uno degli articoli più influenti nella storia della matematica.

Tarski, A. (1936). Der Wahrheitsbegriff in den formalisierten Sprachen. Studia Philosophica, 1, 261–405. — L'indefinibilità della verità aritmetica.

Löb, M.H. (1955). Solution of a Problem of Leon Henkin. Journal of Symbolic Logic, 20(2), 115–118. — Il Teorema di Löb.

Silvano Salvador

Solovay, R.M. (1976). Provability Interpretations of Modal Logic. Israel Journal of Mathematics, 25, 287–304. — Completezza di GL.

Boolos, G., Burgess, J.P. e Jeffrey, R.C. (2007). Computability and Logic (5ª ed.). Cambridge: Cambridge University Press. — Testo di riferimento per logica matematica e calcolabilità.

II. Teoria della calcolabilità e della complessità

Turing, A.M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 42(2), 230–265.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics, 58(2), 345–363.

Rice, H.G. (1953). Classes of Recursively Enumerable Sets and Their Decision Problems. Transactions of the AMS, 74(2), 358–366.

Cook, S.A. (1971). The Complexity of Theorem Proving Procedures. Proceedings STOC, 151–158.

Papadimitriou, C.H. (1994). Computational Complexity. Reading, MA: Addison-Wesley.

Arora, S. e Barak, B. (2009). Computational Complexity: A Modern Approach. Cambridge University Press.

Razborov, A. e Rudich, S. (1994). Natural Proofs. Journal of Computer and System Sciences, 55(1), 24–35.

Silvano Salvador

III. AI simbolica, logica applicata e sistemi esperti

Newell, A. e Simon, H.A. (1976). Computer Science as Empirical Inquiry. Communications of the ACM, 19(3), 113–126.

Robinson, J.A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. Journal of the ACM, 12(1), 23–41.

McCarthy, J. (1963). Situations, Actions, and Causal Laws. Stanford AI Memo 2.

Minsky, M. e Papert, S. (1969). Perceptrons. Cambridge, MA: MIT Press.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Mateo: Morgan Kaufmann.

Pearl, J. (2000). Causality: Models, Reasoning and Inference. Cambridge University Press.

Richardson, M. e Domingos, P. (2006). Markov Logic Networks. Machine Learning, 62(1–2), 107–136.

IV. Connessionismo, reti neurali e deep learning

McCulloch, W.S. e Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5(4), 115–133.

Rosenblatt, F. (1958). The Perceptron. Psychological Review, 65(6), 386–408.

Silvano Salvador

Rumelhart, D.E., Hinton, G.E. e Williams, R.J. (1986). Learning Representations by Back-propagating Errors. Nature, 323, 533–536.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals, and Systems, 2(4), 303–314.

Hornik, K., Stinchcombe, M. e White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. Neural Networks, 2(5), 359–366.

Hochreiter, S. e Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

Krizhevsky, A., Sutskever, I. e Hinton, G.E. (2012). ImageNet Classification with Deep CNNs. NeurIPS 25.

He, K., Zhang, X., Ren, S. e Sun, J. (2016). Deep Residual Learning for Image Recognition. CVPR, 770–778.

Vaswani, A. et al. (2017). Attention Is All You Need. NeurIPS 30.

Srivastava, N. et al. (2014). Dropout. JMLR, 15, 1929–1958.

Ioffe, S. e Szegedy, C. (2015). Batch Normalization. ICML, 448–456.

Kingma, D.P. e Ba, J. (2015). Adam: A Method for Stochastic Optimization. ICLR.

Telgarsky, M. (2016). Benefits of Depth in Neural Networks. COLT, 1517–1539.

Chen, R.T.Q. et al. (2018). Neural Ordinary Differential Equations. NeurIPS 31.

Kaplan, J. et al. (2020). Scaling Laws for Neural Language Models. arXiv:2001.08361.

Silvano Salvador

Su, J. et al. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864.

V. Verifica formale e AI simbolica applicata

Clarke, E.M., Emerson, E.A. e Sistla, A.P. (1986). Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic. TOPLAS, 8(2), 244–263.

Katz, G. et al. (2017). Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. CAV, 97–117.

Gehr, T. et al. (2018). AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation. IEEE S&P.

Cohen, J.M. et al. (2019). Certified Adversarial Robustness via Randomized Smoothing. ICML.

Klein, G. et al. (2009). seL4: Formal Verification of an OS Kernel. SOSP, 207–220.

Leroy, X. (2009). Formal Verification of a Realistic Compiler. CACM, 52(7), 107–115.

Gonthier, G. et al. (2013). A Machine-Checked Proof of the Odd Order Theorem. ITP, 163–179.

Hales, T. et al. (2017). A Formal Proof of the Kepler Conjecture. Forum of Mathematics Pi, 5.

Rocktäschel, T. e Riedel, S. (2017). End-to-End Differentiable Proving. NeurIPS 30.

Silvano Salvador

Trinh, T.H. et al. (2024). Solving Olympiad Geometry without Human Demonstrations. Nature, 625, 476–482.

VI. Teoria statistica dell'apprendimento e fondamenti

Valiant, L.G. (1984). A Theory of the Learnable. CACM, 27(11), 1134–1142.

Vapnik, V.N. (1995). The Nature of Statistical Learning Theory. Springer.

McAllester, D.A. (1999). Some PAC-Bayesian Theorems. Machine Learning, 37(3), 355–363.

Shannon, C.E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27(3–4), 379–423.

Amari, S. (1998). Natural Gradient Works Efficiently in Learning. Neural Computation, 10(2), 251–276.

Rissanen, J. (1978). Modeling by Shortest Data Description. Automatica, 14(5), 465–471.

Wolpert, D.H. (1996). The Lack of A Priori Distinctions Between Learning Algorithms. Neural Computation, 8(7), 1341–1390.

VII. Governance AI, fairness, privacy e allineamento

Dwork, C. et al. (2006). Calibrating Noise to Sensitivity in Private Data Analysis. TCC, 265–284.

Silvano Salvador

Dwork, C. et al. (2012). Fairness Through Awareness. ITCS, 214–226.

Kusner, M. et al. (2017). Counterfactual Fairness. NeurIPS 30.

Abadi, M. et al. (2016). Deep Learning with Differential Privacy. CCS.

Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.

Regolamento (UE) 2024/1689 del Parlamento Europeo e del Consiglio (AI Act). Gazzetta Ufficiale dell'UE.

Shpitser, I. e Pearl, J. (2006). Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models. AAAI, 1219–1226.

VIII. Filosofia della mente e fondamenti dell'AI

Turing, A.M. (1950). Computing Machinery and Intelligence. Mind, 59(236), 433–460.

Searle, J.R. (1980). Minds, Brains, and Programs. Behavioral and Brain Sciences, 3(3), 417–424.

Lucas, J.R. (1961). Minds, Machines and Gödel. Philosophy, 36(137), 112–127.

Penrose, R. (1989). The Emperor's New Mind. Oxford University Press.

Dennett, D.C. (1991). Consciousness Explained. Boston: Little, Brown.

Rumelhart, D.E. e McClelland, J.L. (1986). Parallel Distributed Processing (2 voll.). Cambridge: MIT Press.

Silvano Salvador

von Wright, G.H. (1951). Deontic Logic. Mind, 60(237), 1–15.

Curry, H.B. e Feys, R. (1958). Combinatory Logic, vol. I. Amsterdam: North-Holland.

Howard, W.A. (1980). The Formulae-as-Types Notion of Construction. In: To H.B. Curry: Essays on Combinatory Logic. Academic Press, 479–490.

IX. Interpretabilità e mechanistic interpretability

Elhage, N. et al. (2021). A Mathematical Framework for Transformer Circuits. Transformer Circuits Thread. Anthropic.

Nanda, N. et al. (2023). Progress Measures for Grokking via Mechanistic Interpretability. ICLR.

Conmy, A. et al. (2023). Towards Automated Circuit Discovery for Mechanistic Interpretability. NeurIPS.

Cammarata, N. et al. (2020). Thread: Circuits. Distill.

Wei, J. et al. (2022). Emergent Abilities of Large Language Models. TMLR.

Silvano Salvador